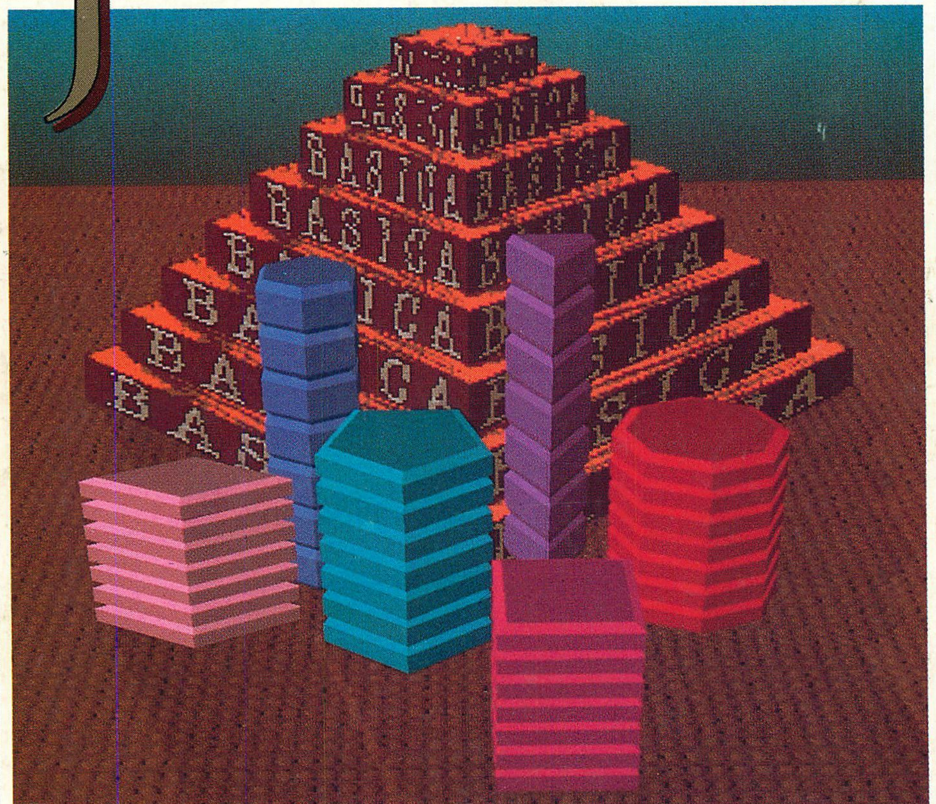
JUNE 1986 VOL. 4, No. 6 $3.95

FOR IBM PERSONAL COMPUTER USERS

PC TECH JOURNAL



SIX NEW SHAPES OF BASIC

Building on BASICA

INSTANT SCREENS

DATA MANAGER: KNOWLEDGEMAN/2

iRMX FOR THE PC

	JRAM-3	Rampage™	AboveBoard™
JLASER	✓		
re-	✓		
	✓	✓	
	✓	✓	
$	269	495	395

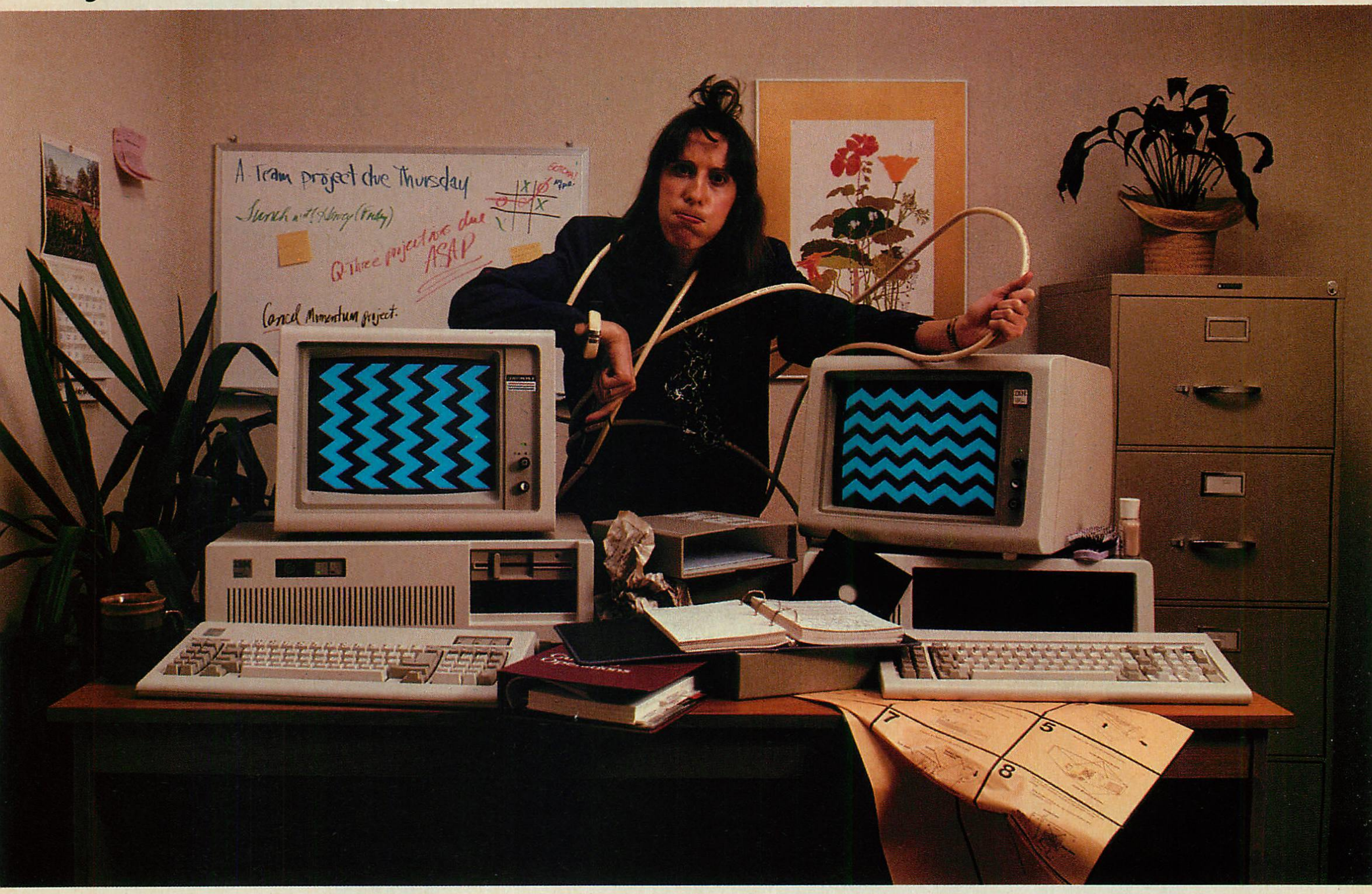
HP LaserJet, Rampage, and Above Board are trademarks of Hewlett-Packard, Canon, AST Research Inc., and Intel Corp., respectively.

Tall Tree Systems
1120 San Antonio Road • Palo Alto, CA 94303 • (415) 964-1980

The text and icons on this page were created on a Composition Technology International PageScan System using a Tall Tree JRAM-3 board and was output on a Corona Laser Printer.

CIRCLE NO. 197 ON READER SERVICE CARD

If you can't share files on your network, you're using the wrong file manager.



Be connected. Btrieve.®

Networks can solve problems. But running a single-user file manager can create new ones: Lost updates. Garbled data. Trashed files.

Btrieve® /N offers *safe* multi-user file management that protects your data when sharing files. And eliminates the need to rewrite your application for networking. Btrieve/N set the file management standard for the industry's most popular LANS: IBM's PC Network, Netware, PC Net and Ether-Series. And now Btrieve/N sets the standard for multi-user systems: XENIX, MultiLink Advanced and others.

Fast. Btrieve/N is fast, too. It's written in assembly language especially for the IBM PC. And based on b-tree file indexing with automatic balancing for access speed that won't degrade as your database grows. With Btrieve/N, your applications always run fast. And users don't waste time waiting.

Automatic file recovery. Btrieve/N provides automatic file recovery after a system crash, so accidents and power failures don't turn into database disasters. Your Btrieve data always comes back intact.

Fully-relational data management. SoftCraft's entire family of products gives you a complete, fully-relational database management system. Rtrieve™/N adds report writing capabilities for generating the reports you need. Xtrieve™/N speeds users through database queries with interactive, on-screen menus—no command language or special syntax.

For professional programmers. Btrieve/N is the fast, reliable answer for all your application development. In any development language—BASIC, Pascal, COBOL, C, FORTRAN and APL. With multikey access to records. Unlimited records per file. Duplicate, modifiable, and segmented keys.

With Btrieve/N, you can develop better network applications. And solve problems, not create new ones.

NO ROYALTIES

Suggested retail prices: Btrieve, $245; Btrieve/N, $595; Xtrieve, $195; Xtrieve/N, $395; Rtrieve, $85; Rtrieve/N, $175. Requires PC-DOS or MS-DOS 1.X, 2.X, or 3.X.

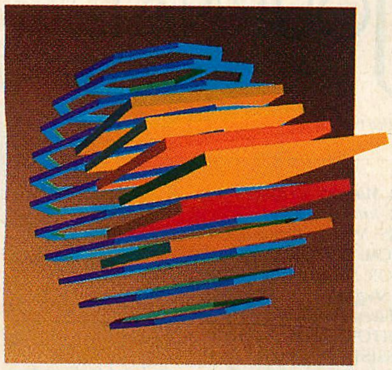
Btrieve is a registered trademark and Xtrieve and Rtrieve are trademarks of SoftCraft Inc.

SoftCraft Inc.
P.O. Box 9802 #917
Austin, Texas 78766
(512) 346-8380 Telex 358 200

CIRCLE NO. 201 ON READER SERVICE CARD

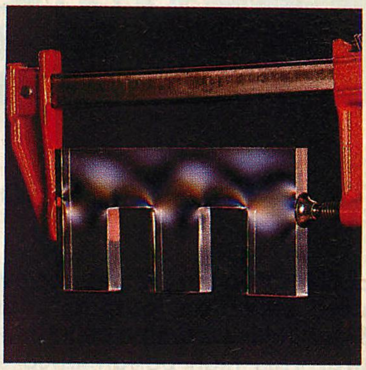
PC TECH JOURNAL

JUNE 1986 VOLUME 4, NUMBER 6



PC Versions of iRMX 126

Mechanical CAD 80



Equation Solving by Formula/One 110

Cover computer graphic · Dov Jacobson

CIRCLE NO. 222 ON READER SERVICE CARD

PC TECH JOURNAL®

VOL. 4, NO. 6

PUBLISHER: *Newton Barrett*
EDITOR: *Will Fastie*

EDITORIAL
MANAGING EDITOR: *Marjory Spraycar*
EXECUTIVE EDITOR: *Julie Anderson*
TECHNICAL EDITORS: *Jeff Duntemann, Caroline Halliday*
ASSOCIATE TECHNICAL EDITOR: *Dan Beale*
SENIOR COPY EDITOR: *Susan Holly*
COPY EDITOR: *Gail Shaffer*
PROOFREADER: *Kathleen Peddicord*
NEW PRODUCTS EDITOR: *Carole Autenzio*
ADMINISTRATIVE ASSISTANT: *Diana Carey*
CONSULTING EDITORS: *Thomas V. Hoffmann, Richard M. Foard*
CONTRIBUTING EDITORS: *Steven Armbrust, Don Awalt, Michael Covington, Ted Forgeron, Augie Hansen, Ted Mirecki, Max Stul Oppenheimer*

ART & PRODUCTION
ART DIRECTOR: *Ina Saltz*
ASSOCIATE ART DIRECTOR: *Sharon Reuter*
ASSISTANT ART DIRECTOR: *Sandra Ray*
ART SECRETARY: *Sabrina Reynolds*
ADVERTISING PRODUCTION MANAGER: *Lisa Franey Ducey*
EDITORIAL PRODUCTION COORDINATOR: *Eve Hinderer*

ADVERTISING SALES
ADVERTISING DIRECTOR: *Rita Burke*
MARKETING COORDINATOR: *Julie Henderson*
ADVERTISING COORDINATOR: *Michele Fischetti*
DISTRICT MANAGERS: *Rosemarie Caruso, Jan Schultz—East Coast; Ted Babr, Bill Bush, Phyllis Egan—West Coast*
ACCOUNT REPRESENTATIVES: *Pat Toohey, Jane Anderson—East Coast; Arlene Braithwaite—Midwest; Pam Sigal, Jane Anderson—West Coast*

CIRCULATION
CIRCULATION MANAGER: *Charles Mast*
CIRCULATION SALES DEVELOPMENT: *Daniel Rosensweig*
MEDIA MANAGER: *Melinda Kendall*

ZIFF-DAVIS PUBLISHING COMPANY
PRESIDENT: *Kenneth H. Koppel*
SENIOR VICE PRESIDENT, Marketing: *Paul Chook*
VICE PRESIDENT, Operations: *Baird Davis*
VICE PRESIDENT, Controller: *John Vlachos*
VICE PRESIDENT, Creative Services: *Herbert Stern*
VICE PRESIDENT, Circulation: *Alicia Marie Ivans*
VICE PRESIDENT, Circulation Services: *James Ramaley*
VICE PRESIDENT, Marketing Services: *Ann Pollak Adelman*
VICE PRESIDENT, Development: *Seth Alpert*
VICE PRESIDENT: *Hugh Tietjen*
BUSINESS MANAGER: *Gary A. Gustafson*
PRODUCTION DIRECTOR: *Walter J. Terlecki*

ZIFF COMMUNICATIONS COMPANY
PRESIDENT: *Philip B. Korsant;* EXECUTIVE VICE PRESIDENT: *James D. Dunning, Jr.;* SENIOR VICE PRESIDENTS: *Philip Sine, Kenneth H. Koppel;* VICE PRESIDENTS: *Laurence Usdin, William L. Phillips, J. Malcolm Morris, Steven C. Feinman;* TREASURER: *Selwyn Taubman;* SECRETARY: *Bertram A. Abrams*

EDITORIAL OFFICE
PC TECH JOURNAL, The World Trade Center, Suite 211, Baltimore, MD 21202. 301/576-0770. FAX (group 3): 301/576-9603. MCIMail: PCTECH. PCTECHline: 301/576-PCTJ. Telex: 6502565932 MCI.

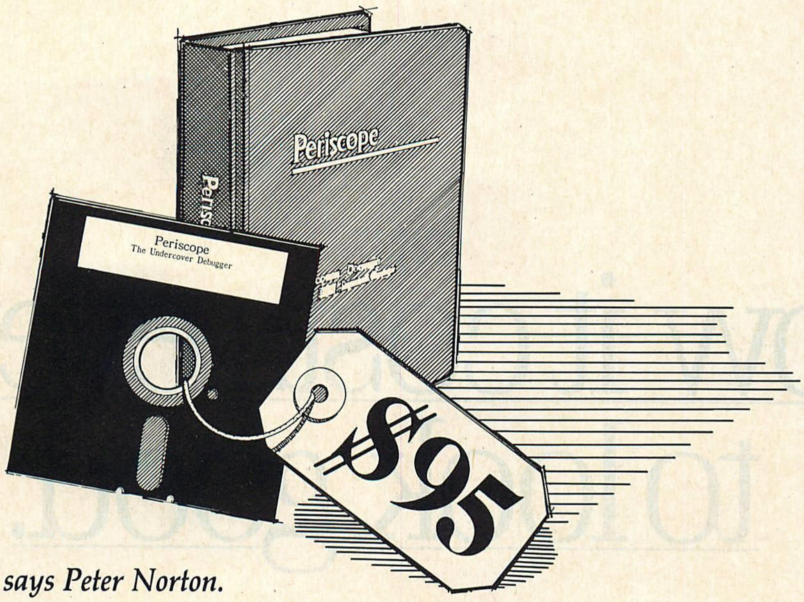
ADVERTISING OFFICES
(East Coast/Midwest) One Park Ave., New York, NY 10016. 212/503-5185. (West Coast) 3460 Wilshire Blvd., Los Angeles, CA 90010. 213/387-2100; 11 Davis Drive, Belmont, CA 94002. 415/598-2290.

SUBSCRIPTION INQUIRIES
PC TECH JOURNAL, P.O. Box 2968, Boulder, CO 80321. Subscription service: 800/525-0643, 303/447-9330. Back issues: send $7/copy ($8 outside U.S.) to Ziff-Davis Publishing, One Park Ave., 4th floor, New York, NY 10016.

PC TECH JOURNAL (ISSN 0738-0194) is published monthly, $29.97 for one year, $52.97 for two years, $69.97 for three years. Additional postage $6 for Canada & Foreign by Ziff-Davis Publishing Company, One Park Avenue, New York, NY 10016. Second-Class Postage paid at New York, NY and at additional mailing offices. POSTMASTER: Send address changes or subscription inquiries to P.O. Box 2968, Boulder, CO 80321.

BPA

Save On New Model of Periscope



"A marvelous debugging tool..." says Peter Norton.

SUMMER SALE.
You don't have to spend thousands, or even hundreds, of dollars to buy a high-performance debugger. Through August 31, the new software-only Periscope II-X is yours for only $95! Periscope II, normally $145, is only $125!

NEW MODEL.
Periscope I includes both a break-out switch for "spontaneous" debugging and a RAM board to protect the debugger from programs that overwrite memory. Periscope II includes a break-out switch that doesn't require an extra slot to install. And the new Periscope II-X is Periscope II without the break-out switch. So you can now get Periscope with no hardware at all!

NEW RELEASE.
The new version 2.1 enhances all models with many of the features users have requested. Call for details.

CHECK OUT THE REVIEWS.

- *"Periscope strikes a balance of power and features that will please serious PC programmers."* — Jeff Duntemann, PRODUCT OF THE MONTH, *PC Tech Journal*, 1/86
- *"This symbolic debugger, with its breakout button and powerful command options, stands unrivaled for its flexibility . . . Periscope's diverse features, affordable price, and portability place it in a class by itself."* —Ward Christensen, "Breaking out with Periscope," *PC Tech Journal*, 3/86
- *"Periscope represents the finest software debugger available in its class."* — Andrew Fried, *Computer Shopper*, 4/86

Other reviews have appeared in *Computer Language* (3/86), the Boston Computer Society's PC Report (1&2/85), and *Programmer's Journal* (Vol. 3, No. 1).

I'M CONVINCED PERISCOPE IS FOR ME!

Please send me the following:

Periscope II-X ____ @ $95* ....____

Periscope II with Switch ____ @ $125* ....____

Periscope I with Board & Switch ____ @ $295..........____

Add Shipping Cost (see below) .....____

Total Order Amount..............____

Shipping-$3 UPS; $5 COD; $8 Canada; $24 Foreign

We accept qualified PO's in the U.S.

*Prices good through August 31, 1986

Ship to:

Name ____

Company ____

Address ____

City/State/Zip ____

Telephone ____

Payment: ☐ Check enclosed ☐ COD ☐ MC ☐ Visa

Card No. ____

Expires ____

Signature ____

PERISCOPE REQUIRES an IBM PC, XT, AT, Compaq, or other close compatible; 128K RAM; DOS 2.0 or later; one disk drive; and an 80-column monitor.

Take advantage of the low sale prices while they last . . . Order your Periscope today! To order, clip the coupon, or call TOLL-FREE

800/722-7006

30-Day Money-Back Guarantee

Get Your Programs Up and Running;

UP PERISCOPE!

Data Base Decisions
14 Bonnie Lane
Atlanta, GA 30328
404/256-3860

CIRCLE NO. 215 ON READER SERVICE CARD

Now it costs even less to look good.

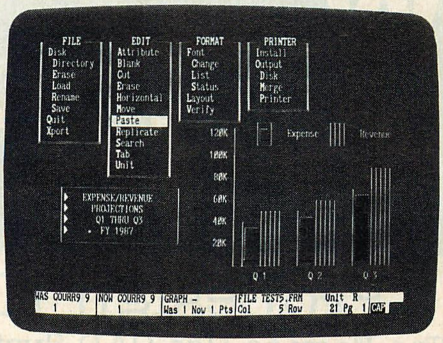
Wordstar 2000 is a trademark of MicroPro International Corp. Epson is a trademark of Epson America, Inc.

The Cordata Desktop Printshop™ has always been the best value in PC-compatible laser printers.

With speed and flexibility that the others can't match.

For example, we put 38 type fonts on diskette, not expensive cartridges. And let you mix up to 32 fonts in a single document. Automatically.

A high-speed graphics interface transfers data up to 30 times faster than Hewlett-Packard's LaserJet+, and because most Desktop Printshop features are implemented in software, they're easy to update.



And because Epson™ emulation is standard, the Desktop Printshop has a ready-to-run solution for many programs that other laser printers can't use.

Not to mention being easier.

But now the Desktop Printshop is an even better value.

A new forms design software program comes with the Desktop Printshop that lets you create documents quickly—and easily—using pull-down menus, not complicated commands.

We've also included the industry-acclaimed Wordstar 2000™ word processing program. So you can start producing near-typeset quality memos, reports, and correspondence right away.

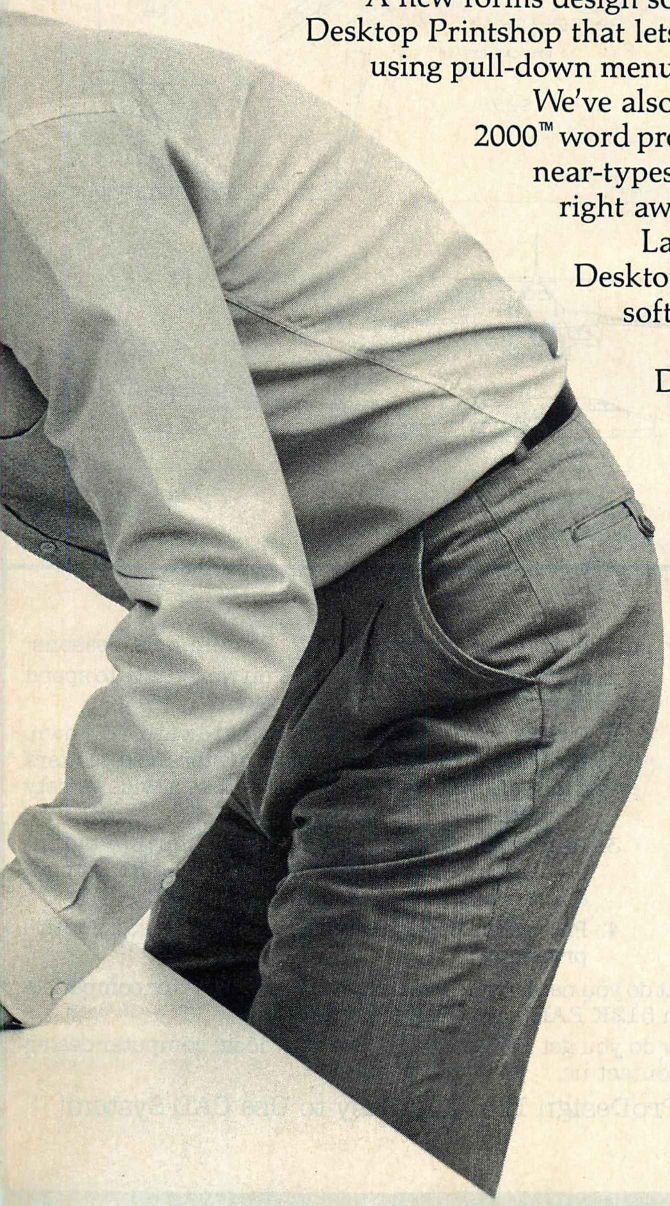
Last, but not least, we've lowered the price of the Desktop Printshop to $2995, including forms design software, Wordstar 2000 and 38 fonts on diskette.

So don't settle for just any laser printer.

Demand the Desktop Printshop.

All it takes is a call to Cordata at: (800) 621-6746 (in CA: (805) 495-5800; in Canada: (604) 984-0641) for the name of your nearest Desktop Printshop dealer.

After all, how can you afford *not* to look your best?



The Desktop Printshop.™

CORDATA, 275 E. Hillcrest Drive, Thousand Oaks, CA 91360
TELEX 650-2696270; in Europe: Holland 032-18111

CIRCLE NO. 117 ON READER SERVICE CARD

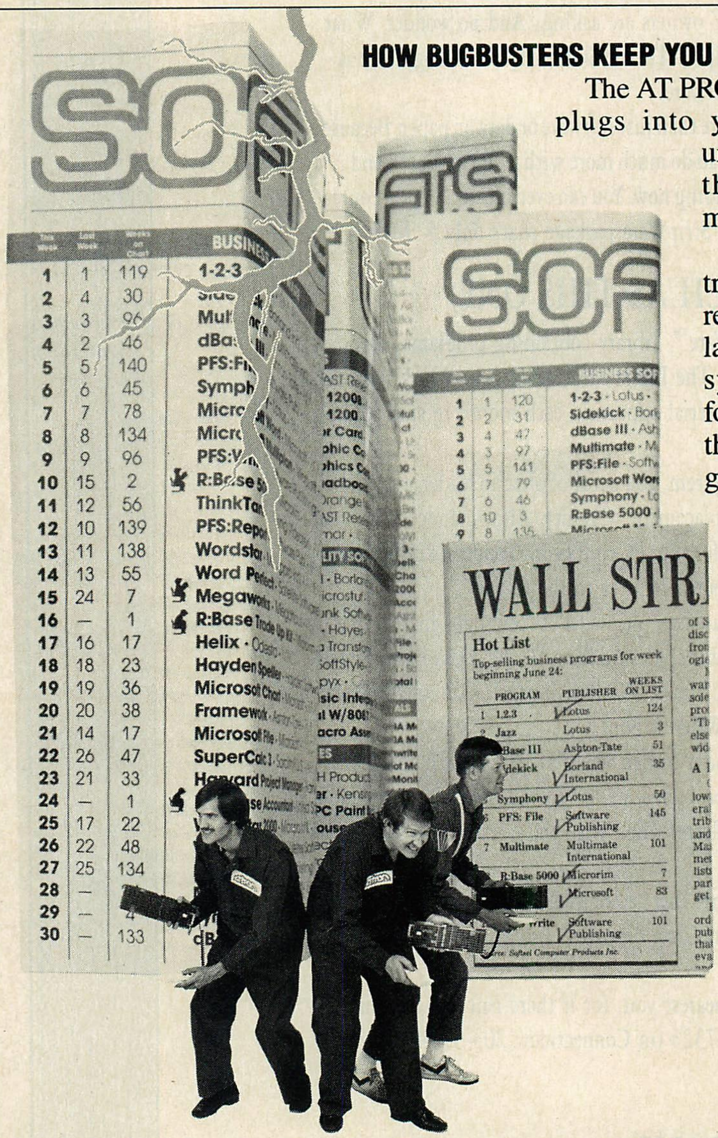
cordata

formerly corona data systems, inc.

ATRON BUGBUSTERS GREASE BORLAND LIGHTNING

"If I were starting a software company again, from scratch, Atron's AT PROBE™ would be among my very first investments. Without Atron's hardware-assisted, software debugging technology, the flash of Turbo Lightning™ would be a light-year away."

Philippe Kahn, President, Borland



HOW BORLAND DOES SO MUCH, SO WELL, SO FAST

We asked Borland International president Philippe Kahn to share his secrets for rapidly taking a good idea and turning it into rock-solid reality. How does the Borland team do so much, so well, so fast?

He begins, "I remember when Atron used the June 24, 1985 *Wall Street Journal* chart of top-selling software in an ad." [Note: At that time, seven of the top ten software packages were created by Atron customers; it's now now nine out of ten.] "SideKick was number four, and I let Atron quote me in saying that there wouldn't have been a SideKick without Atron's hardware-assisted debuggers.

"You might say lightning has literally struck again. Turbo Lightning made number four on *SoftSel's Hotlist* within weeks of its introduction! And again, I say we couldn't have done it without Atron debugging technology.

"Cleverly written code is, by definition tight, recursive, and terribly complex," he continues. "Without the ability to externally track the execution of this code, competent debugging becomes very nearly impossible."

Concludes Philippe, "And after Turbo Lightning was solid and reliable, Atron tuning software turned our Probes into performance analyzers. How do you think we *greased* our lightning?"

Philippe, along with a couple million or so of your satisfied customers, we say congratulations on yet another best-selling product. We can't wait to see what awesomely useful technology will come shooting out of Borland International next.

HOW BUGBUSTERS KEEP YOU FROM GETTING SLIMED

The AT PROBE is a circuit board that plugs into your PC/AT. It has an umbilical which plugs into the 80287 socket and monitors all 80286 activity.

Since AT PROBE can trace program execution in real time, and display the last 2048 memory cycles in symbolic or source-code form, you can easily answer the questions: "How did I get here?" and "What *are* those silly interrupts doing?"

It can solve *spooky* debugging problems. Like finding where your program overwrites memory or I/O - impossible with software debuggers.

You can even do source-level debugging in your favorite language, like C, Pascal or assembler. And after your application is debugged, the AT PROBE's performance measurement software can isolate performance bottlenecks.

Finally, the AT PROBE has its own 1-MByte of memory. Hidden and write-protected. How else could you develop that really large program, where the symbol table would otherwise take up most of memory.

LOOK AT IT THIS WAY.

History shows that non-Atron customers don't stand a very good chance of making the Top Ten list. Lightning *really does* have a way of striking twice!

The PC PROBE™ is $1595 and the AT PROBE is $2495. So call Atron today. You can be busting some really scarey bugs tomorrow. And maybe, just like Borland, you can also bust some records.

Atron
THE DEBUGGER COMPANY
20665 Fourth Street • Saratoga, CA 95070 • 408/741-5900

Bigger, Smaller, Faster, Slower

A raft of new PCs from Compaq and IBM keep this editor's blood flowing.

Heaven. Editors of computer magazines, I think, live for the introduction of new machines. There is nothing I like more than sitting in my office surrounded by many computers of all shapes, sizes, colors, and descriptions. Sometimes I even use them.

We have really struck pay dirt lately with the RT/PC, the Compaq Portable II (see Product of the Month, this issue, p. 31), new XT models and options, a new AT model, and, finally, IBM's long-awaited and long-predicted laptop, the PC Convertible (see Tech Releases, this issue, p. 32). Compaq has been busy the past two years developing its 286 machines, but IBM has been relatively silent. Suddenly, an explosion of hardware and software from IBM and some excitement from Compaq shatters our complacency and editorial schedule.

The big news is, of course, IBM's announcement of the PC Convertible, the first *true* portable from Big Blue. Characterized by an LCD display no better than any other, limited memory (given the price), and sluggish disk performance, the significance of this machine must be found elsewhere.

Indeed, the Convertible has some interesting aspects. First is the reported reason the machine lost out in bidding for the IRS contract: 3½-inch diskette drives. Concern over data interchange between machines with 5½-inch drives is mitigated by IBM-provided 3½-inch drives for the PC family and by the fact that data interchange requirements are probably not as great a problem as everyone thinks. What the IRS overlooked was not only the greater capacity of the little guys (720KB), but also the inherent *reliability* of the media. The smaller diskettes afford a much higher level of protection for the media with their rigid jacket and retracting access cover. For a portable computer that will be subjected to extremes of use, wear and tear on the media is a very important consideration.

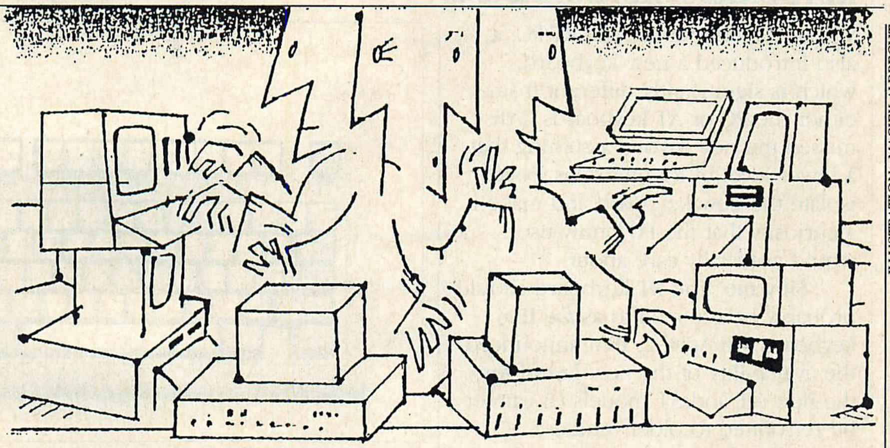


ILLUSTRATION • MACIEK ALBRECHT

A second important characteristic of the PC Convertible is the reason for its name. The user can detach the LCD display and attach a regular monitor (although an additional adapter is required for such use). IBM touts the machine as one that can be used both as a portable computer and as a desktop. If a new, more desirable display becomes available, IBM can integrate the improved device into the manufacturing process. The customer pays for the detachability to begin with, but long-term benefits include lower costs associated with efficient manufacturing. The customer also may be able to retrofit to a better display, avoiding early obsolescence of the machine.

The single most important part of the PC Convertible is the set of IBM-built integrated circuits that replace most of the off-the-shelf logic found in the desktop PC. The Convertible is the medium with which IBM puts the Far East on notice: if necessary, an 8088-based machine (like the PC or XT) can be built here and, obviously, be competitive. Not only that, it can be built almost completely without third-party parts. Although 80286-based machines will become the most sought-after desktops, the basic PC architecture is viable, usable, and, as IBM shows us in the Convertible, economical.

That leads to the price of the Convertible. At first glance, $1,995 for a two-drive unit with an IBM label seems reasonable. The actual price, however, is $2,375 because today's software demands more than the standard 256KB of RAM. Worse, that extra $380 expands the machine to only 512KB, the maximum configuration. A number of popular software packages simply cannot function well (or at all) without 640KB.

Possessed of many nice features, a reasonable configuration, and the IBM logo, the PC Convertible should become a popular machine. It is not, however, what all of us want: a standard PC squashed down to the size of a book.

BIGGER AND FASTER

The PC Convertible was just one of a long list of announcements from IBM in April. IBM has restructured its entire PC/XT/AT product line, including derivatives such as the 3270- and the -/370, with new system boards, faster AT performance, and bigger disks.

An eight-megahertz AT is nothing more than a catch-up machine. IBM is surprisingly late with this development, and it may have cost the company a significant early share of the 286-based market. It is attempting to recapture a share with a $5,295 base price for a machine with 512KB of memory (why

on earth 640 is not the standard escapes me still), the 8-MHz clock, and a 30MB fixed disk. Strangely, the current 6-MHz AT remains, in two configurations. The first is exactly equivalent to the new model except for the clock speed; it lists at $5,295, the same price as the new one—inexplicable, to say the least. The original AT with its 20MB disk was reduced $900 and is now $4,895.

The new XT is more interesting. A machine with one half-height diskette drive (that's right, half-size), *20MB* of fixed disk, and 512KB of memory is $2,895, a price that is quite competitive after discount. The new XT system board supports memory expansion up to 640KB. For situations in which processor performance is not critical, the new XT should serve well and provide ample opportunity for expansion with its five available slots.

The new XT design also provides for an internally mounted, half-height 3½-inch diskette drive to provide for data interchange with the Convertible or just for larger capacity diskettes.

From a machine standpoint, the IBM announcements were significant. The new models and the price changes give IBM a much more balanced product family. Some problems linger on, and some of IBM's decisions seem peculiar, but notwithstanding buyer confusion, IBM now offers a better set of choices for its customers.

KEYBOARDS, KEYBOARDS, ...

When IBM announced the RT/PC, it also introduced a new keyboard, which is significantly different from either the PC or AT keyboards. I dismissed the new layout, assuming that a lower level of RT unit sales would isolate the new keyboard and make it a curiosity that the PC-family user would not really care about.

Silly me. The RT keyboard should probably be referred to as *the* IBM keyboard: on April 2, IBM announced the availability of the new layout for the new XT and AT models (it cannot be retrofitted to older models).

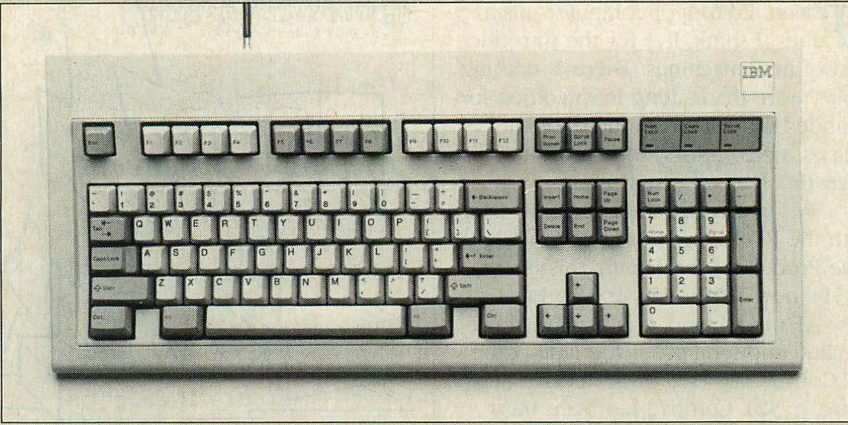
PC Tech Journal will examine the keyboard in detail in a forthcoming review, but a number of observations can be made at this juncture.

First, IBM is very brave. Some of the changes in the new keyboard will frustrate users. Software developers will have bad dreams about designing products to work on a variety of layouts or providing keyboard installation routines. I expect many to scream in anguish at Big Blue.

The two most noticeable changes are the placement of 12 function keys along the top of the keyboard (where they belong, after all) and the presence of a separate set of cursor keys. Most users will find these changes satisfactory and long overdue. The placement of keys such as Ins, Del, Home, End, PgUp, and PgDn is more questionable. IBM hedged its bets: the numeric keypad still carries the old legends and can be used, if desired.

IBM returned the Esc key to its home on the upper left of the keyboard (many questioned why it was ever moved). But instead of nestling it next to the 1 key, it floats separately next to the function keys—a perfect solution. Esc is now where it belongs and is easy to hit without looking.

Another improvement is labeling keys such as Tab and Backspace with



The new IBM "universal" keyboard (PC/AT version)

words in addition to the arrows that marked them before. The new keyboard clearly identifies *every* key.

The big problem with this keyboard is that the Ctrl key has migrated. A surprising number of programs use control keys for cursor movement, and an even larger number of users have fingers that remember. WordStar devotees: look out! IBM will argue that the original keyboard is still available and that most programs using control characters can use the regular cursor keys as well. No matter. The Ctrl key is still in the wrong place (in fact, in two wrong places). Worse, it is now where the Alt key used to be to the left of the Space Bar (and in a corresponding position on the right). Alt moved to two new places flanking the Space Bar.

Why did the Ctrl key move? To make room for the CapsLock key, which was repositioned to where it is found on typewriters. That is a reasonable move; the Ctrl key placement, however, is not. After 15 years of keyboarding, it will take me a long time to get used to a Ctrl key I cannot find.

The final problem is the Enter key. It has shrunk from the large, double-row, L-shaped key of the AT to a single-row, double-width key. The apparent reason for this is to make room for yet another placement of the backslash/vertical bar key.

If that were not enough, IBM also has given us a different keyboard layout on the PC Convertible. Some compromises are to be expected on smaller machines, especially when the prime concern is proper key spacing and full-stroke keys. I can even accept the keypunch-style embedding of the numeric keys on the alpha keys and their activation with a PC*jr*-like Fn key. However, IBM made Home, End, PgUp, and PgDn functions of the cursor keys, a move bound to frustrate users of most word processors. Also, because the function keys have very little adjoining real estate, templates are hard to fit.

We can hardly object to the evolution of keyboards. However, IBM is a company with vast experience building keyboards and should, by now, have a substantial base of human factors knowledge from which to build. In terms of key placement, IBM's experience is still not showing.

—*WF*

YOUR COMPUTER LANGUAGE IS QUIETLY BREEDING REAL BATS IN YOUR BELFRY.

WE'RE OUT TO SAVE ONE MILLION FRUSTRATED PROGRAMMERS

You're on a roll, really pumped, writing the best code you have ever written and then—AAARGHHH!

Freeze dried in your tracks because the language you're using just won't let you achieve what you can conceive.

And you wanted to be a programmer.

So your choices are:

1) write around the problem by creating six pages of emetic code...

2) leave out that incredible idea that really puts your stamp of excellence on this program or...

3) get yourself a world class headache (or a stroke) by dropping into assembler.

Whatever you choose, by now you feel the language is out to get you—because it is.

Sure, no language is perfect, but you have to wonder, "Am I getting all I deserve?"

And, like money, you'll never have enough.

Pretty dismal, huh?

We thought so, too.

So we did something about it.

We call it CLARION.

You'll call it incredible.

With CLARION you can write, compile, run and debug complex applications in a New York afternoon.

Even if you're in Savannah.

It gives you the power and speed to create screens, windows and reports of such richness and clarity you would never attempt them with any other language.

Because YOU would have to write the code.

With CLARION you simply design the screens using our SCREENER utility and then CLARION writes the source code AND compiles it for you.

In seconds.

Likewise, you can use REPORTER to create reports. Remember, only CLARION can recompile and display a screen or report layout for modification.

And with no time wasted.

All the power and facilities you need to write great programs, faster than you ever dreamed of.

Programs that are easy to use.

Programs that are a pleasure to write.

And to you that means true satisfaction.

You've coveted those nifty pop-up help windows some major applications feature. But you can't afford the time and energy it takes to write them into your programs.

That's the way it used to be.

So we fixed that, too.

CLARION HELPER is an interactive utility that let's you design the most effective pop-up help screens that you can imagine. And they're "context sensitive," meaning you can have help for every field in your application.

Unlike the other micro languages, CLARION provides declarations, procedures and functions to process dates, strings, screens, reports, indexed files, DOS files and memory tables.

Imagine making source program changes with the CLARION EDITOR. A single keystroke terminates the EDITOR, loads the COMPILER, compiles the program, loads the PROCESSOR and executes the program. It's that easy!

Our data management capabilities are phenomenal. CLARION files permit any number of composite keys which are updated dynamically.

A file may have as many keys as it needs. Each key may be composed of any fields in any order. And key files are updated whenever the value of the key changes.

Like SCREENER and REPORTER, CLARION's FILER utility also has a piece of the CLARION COMPILER. To create a new file, you name the Source Module. Then you name the Statement Label of a file structure within it.

FILER will also automatically rebuild existing files to match a changed file structure. It creates a new record for every existing record, copying the existing fields and initializing new ones.

Sounds pretty complicated, huh?

Not with CLARION's documentation and on-line help screens. If you are currently competent in Basic, Pascal or "C" you can be writing CLARION applications in a day. In two days you won't believe the eloquence of your CLARION programs.

Okay, now for the best part of all. You can say it in CLARION for $295.00—complete. All you need is an IBM® PC, XT, AT or true compatible, with 320 KB of memory and a hard disk drive.

And we'll allow a full 30 day evaluation period. If you're not satisfied with CLARION, simply return it in its original condition for a full refund.

If you're not ready to take advantage of this no-risk opportunity, ask for our detailed 16 page color brochure. It vividly illustrates the elegance of CLARION. Consider it a preview of programming in the fast lane.

Either way, the call's a freebie.

SAY IT IN CLARION™

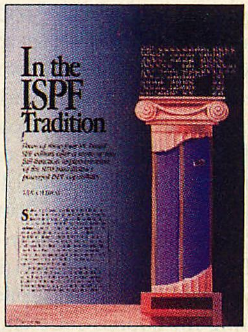
DEPT. A1TJ/1

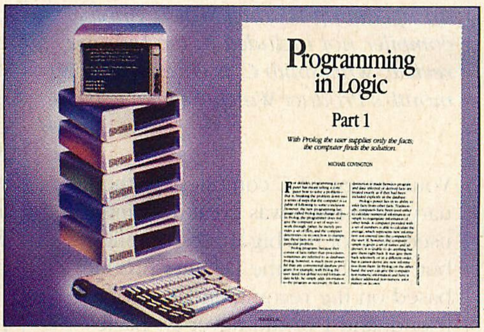
1-800-354-5444

VISA MasterCard

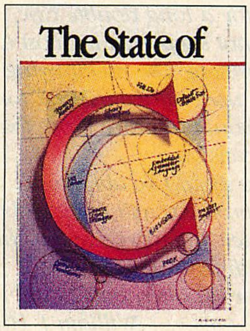
CIRCLE NO. 105 ON READER SERVICE CARD

BARRINGTON SYSTEMS, INC. • P.O. BOX 5580 • POMPANO BEACH, FL 33074 • 305/785-4555


In the ISPF Tradition


Programming in Logic

Part 1


The State of C

EDITOR MISSING

Rudy S. Spraycar's review of ISPF editors for the PC, "In the ISPF Tradition" (March 1986, p. 113), was an excellent introduction to the facilities of IBM's mainframe ISPF. I am a mainframe ISPF user. My transition from mainframe to personal computer was made much simpler by the availability of an ISPF editor for the PC. However, the editor I use, IBM's EZ-VU Editor, was not among the products reviewed.

A review cannot be unbiased if it is not complete. If you do not review all of the available products, you should state why you have not done so. If products become available too late to be reviewed, they should be noted at the end of an article.

I enjoy your magazine, but such oversights could change my opinion.

Joseph R. Smith
Kingston, NY

Thank you for pointing out our oversight. We plan to publish a review of EZ-VU in our Product Watch department in an upcoming issue.

—WF

C NOTES

We at Datalight were pleased with the C compiler review in your January 1986 issue ("The State of C," William J. Hunt, p. 82). We would, however, like to note changes to Datalight C that have been made since your review. We are now delivering a full-featured MAKE program and a compiler control program, and we have introduced our large memory model compiler.

One point though: Datalight C is usable on a dual floppy-disk system. The compiler and related programs fit on one disk, leaving the second completely free for development.

Keep up the good work!

Roy L. Sherrill III
President
Datalight

Mr. Sherrill suggests that Datalight C is usable on a system that has two floppy disk drives (and no hard disk) because the compiler and related files fit onto one floppy diskette. However, in order to avoid switching disks as the user moves from editing through compiling and linking to execution steps, he must keep on-line his editor, linker, C libraries, and some essential DOS files, as well as the compiler. For the Datalight product, those files require 413KB—which is more than the 354KB available on a 360KB floppy diskette. A RAM disk of 100KB to 300KB or a hard disk solves the problem and is a good idea, whatever C compiler is used.

—William J. Hunt

Congratulations to William J. Hunt for his review of C compilers. This is one of the most comprehensive reviews I have seen on the subject. I am sure it will go a long way toward helping serious programmers make better decisions about the software they choose.

Mr. Hunt's evaluation of our compiler is factual and on target. However, the tables that were included with the article had a few omissions. Table 1 omitted the fact that Computer Innovations' Optimizing C86 does support medium memory models. Also, CI C86 does have a cross compiler hosted on VAX/VMS. Table 2 omitted the fact that C86 Optimizing has prologue and epilogue files for assembly language.

George Eberhardt
President
Computer Innovations, Inc.
Tinton Falls, NJ

Mr. Eberhardt points out that his compiler supports the medium memory model. This capability was not documented in the version that was reviewed (2.3a). The manual does provide some information about assembly language, but it offers no examples.

—William J. Hunt

With regard to "The State of C," I wonder if you can help with answers to a few questions. Which companies provide multiple floating-point libraries? Does C have the ability to print a double precision value to 16 (or 17) places of accuracy? How does one determine if a C product (such as a floating-point library) will work with the Datalight compiler? Finally, was the MIX C compiler omitted on purpose?

Dexter Shoultz
Walnut, CA

Some companies that offer floating-point libraries are as follows: Microsoft provides three, one works with in-line 8087 instructions; one works with calls to library functions for float, add, subtract, and other operations; and one implements a faster, less accurate version of floating-point arithmetic. Mark Williams C has different libraries for use with its linker and the Microsoft linker. Lattice C now provides a choice of in-line 8087 floating-point instructions or library calls; both options use a single library. The DeSmet compiler includes software and 8087 float libraries.

The number of libraries is a poor measure of functionality. If you need good floating-point support, you might ask compiler vendors these questions: Does your product represent float data the same way whether software or 8087 floating point is used? Can a single .EXE file work when no 8087 chip is present, but use the 8087 efficiently if it is present? Is floating-point implemented to meet the IEEE standard? Is a faster software float format available for applications where lower precision than IEEE and standard C specify?

Float values certainly can be printed in a format with more than 16 digits. However, most IBM PC C compilers implement the float data type as a 32-bit value and the double data type as a 64-bit value. This gives about 6 digits of precision for float and 15

digits for double. The ANSI standard for C does allow for a long double type; however, it is not supported by most C compilers at present. The 80-bit value that would be used on the IBM PC for long double would produce more than 16 digits of precision.

C vendors should be able to say whether their products will work with a specific compiler. If the product has not been tested with the compiler, you should look for an alternative.

—William J. Hunt

The MIX C compiler will be covered in an upcoming Product Watch. Another compiler not included in the January review, Whitesmith C, is covered in this month's Product Watch. See page 195.)

—DB

Your review of C compilers in the January 1986 issue was great. I formerly used one of the big three compilers, but I purchased the DeSmet product based on the recommendation of a friend, and I would never go back.

The DeSmet compiler is fast and produces fairly efficient code. Its functions, although few, do exactly what they are supposed to do with no side effects. The dynamic memory allocation can be controlled by the compiler (with no bugs) or the compiler can give me pointers and let me control it all. Probably its most important feature is that it will generate full assembly language code, not some pseudo listing. This tells me exactly how the compiler is interpreting my source code and lets me tighten up the program.

Douglas Hill
St. Louis, MO

A.D.A. ADDITIONS

I am the author of A.D.A. Prolog systems, and I was impressed with Michael Covington's review of the products, particularly his suite of benchmarks (see "Programming in Logic," December 1985, p. 82; January 1986, p.145). However, the review process is always a little unsatisfying. It seems as if performance considerations that would become obvious to a user during prolonged use are left undiscovered.

Substantial rewriting has taken place at A.D.A. since the the time of the review and many new features have been added to our Prologs. The benchmarks that do arithmetic were particularly unfortunate. The A.D.A. Prologs compute integers as 32-bit, long signed quantities. At the time of the review, the dedicated pointer representation of an integer was not used, which meant that unique objects representing each integer had to be created, and this takes time. Long arithmetic takes time. To wit, the revised benchmarks would be 90 LIPS for SPEED1.PRO, 25 LIPS for SPEED101.PRO, 1,551 recursions for RECURSE1.PRO, and 1,792 recursions for RECURSE2.PRO.

The garbage collection pause to which Mr. Covington referred was, in fact, a bug. The product is now totally concurrent and no pauses occur. The price of the VML Prolog is $200, not $300 (as of June 1985). All A.D.A. Prolog systems are large model products, and PD Prolog is able to access the full memory of the machine.

All A.D.A. systems now incorporate screen graphics primitives. The more advanced versions blend Prolog with LISP. Where else can you find a Prolog with multidimensional arrays, global and local variables that can bypass instantiation and backtracking, and a quoted variable class? Persons doing pattern analysis had better have arrays.

FOR C AND UNIX

WINDOWS FOR DATA™

PC DOS/UNIX COMPATIBILITY

MENUS WINDOWS DATA ENTRY

PROFESSIONAL DEVELOPERS, here is the front end package you've been waiting for, the ONE that does the hard jobs that others can't — we **guarantee** it. Makes standard display and entry tasks easy. Reliable. Compact. Portable.

MENUS: Build multi-level menus in the format of your choosing: Lotus 1-2-3, Macintosh, or any style you might select. Items can call sub-menus, data-entry windows, or action functions. The menu system is completely flexible.

WINDOWS: WFD is built upon and includes **Windows for C,** the windowing system rated #1 in PC Tech Journal (William Hunt, July 1985). WFC now has more features than ever: unlimited windows and files, pop-ups, fast screen changes, window names, horizontal and vertical scrolling, logical video attributes, highlighting, support for the EGA, off-screen updating, formatted output, word-wrap and auto scroll, print windows, read and write functions, and keyboard input with subroutine execution during waits.

DATA ENTRY: The most complete and flexible data entry system on the market. Pop-up data-entry windows, field types for all C data types, plus special decimal, date, and time fields, full-featured field editing, auto conversion to and from strings for all field types, input masks or "pictures," protected text, system and user-supplied validation functions, range-checking, scrollable context-sensitive help, required and must-fill fields, passwords, programmer definable keys, field types, and mask functions, date and time utilities, and string utilities. Read field by field or auto-read all fields. Branch and nest window forms.

FLEXIBLE

As many possibilities as Vermont in June.

WINDOWS FOR DATA HAS UNPRECEDENTED FLEXIBILITY.

Virtually every capability and feature of WFD can be modified to meet special needs. All key-invoked data-entry functions can be assigned to keys of your choosing; and you can add your own functions to the key assignment table. This same flexibility exists for the input masks used to control data entry. Install your own validation functions. You can even define new field types and add them to the system. You do not need source code to take advantage of the flexibility of WFD, but **full source is available.**

MICROSOFT WINDOWS COMPATIBILITY

is automatic. **Windows for Data** detects the presence of MS Windows **(and IBM's TopView)** and follows the rules required for full compatibility. No need for special code, complex interfaces, or expensive toolkits.

EASE OF USE

Ease of use comes first and foremost from basic design and implementation. WFD is not just a library of functions, but an integrated system for menus, windows, and data entry.

We make the system easy to learn by explaining each major application in step-by-step detail. WFD is documented for the professional. Six hundred pages of documentation in a full-size, high-quality binder. Numerous tutorials and demonstration programs are provided. Nearly two hundred functions are documented individually, to UNIX standards.

RELIABLE

As free from bugs as Vermont in January.

For its basic input and output, WFD uses the library of **Windows for C,** a mature product that has earned a reputation for extreme reliability. WFD has been through its "shakedown", and the few bugs that turned up have been corrected. We promise to quickly respond to any further bug problems you may encounter.

THE MEMORY FILE FACILITY

of windows for C is more flexible and memory efficient than the "virtual screen" systems of other windowing packages. Memory usage adjusts to the amount of text in files. **No waste space!** Build files of any length and width from disk, code, or communications input. Retrieve, replace, add, and scroll file lines. Open windows at any point in a memory file. Scroll windows horizontally or vertically.

UNIX, DOS, OR BOTH

WFC and WFD provide source code compatibility between PCDOS and UNIX. Programs written for one operating system will compile and run on the other with only minor changes.

UNIX developers, now you can put advanced windows, menus, and data entry features common to the PC world in your UNIX programs.

PRAISE FROM USERS

"WFD is the best programming tool I've ever used. It's the most flexible I've seen. Whenever I've wanted to do something, I've been able to find a way."
*Steven Weiss, Stratford Systems (18 yrs; 1 yr)**

"WFC is the standard by which we judge all other C utilities. The most helpful tool we've ever acquired. Absolutely easy to use. Very tight code."
*James Baker, Mathew Bender (7 yrs; 4 yrs)**

"Especially compared to **Panel,** *I love* **Windows for Data.** *Your documentation is great."*
*Don Heinmeller, Law Software (10 yrs; 4 mo)**

"The documentation lets you get up and running fast. I integrated help routines into existing educational programs in a day and a half."
Richard Rovinelli, Educational Services (17 yrs; 1 yr)

**(programming experience; C experience)*

If you are tired of screen utilities that are hard to use and limited in capability, you owe it to yourself and your programs to try WINDOWS FOR DATA.

OUR CHALLENGE AND GUARANTEE

If you have an application where no other tool can do the job, try **Windows for Data.** If it doesn't help you solve your problem, RETURN FOR A FULL REFUND. YOU MUST BE SATISFIED.

	WINDOWS FOR DATA	WINDOWS FOR C
PC DOS*	$295	$195
XENIX-286	$595	$395
UNIX	CALL	CALL

*For all popular C compilers;
No royalties for DOS

Vermont Creative Software
21 Elm Ave.
Richford, VT 05476
802-848-7738, ext. 21

MasterCard & Visa Accepted. Shipping $3.50
VT residents add 4% tax.

Trademarks: Panel, Roundhill Computer Systems; Microsoft, (registered) Microsoft Corp., TopView, IBM.

The most surprising thing you can do with A.D.A. Prolog is use it as a data retrieval system. The system was designed for 1GB (gigabyte) of virtual memory per file, with an unlimited number of files. These limitations are constrained by DOS; however, when they are lifted, they will become actual. Atom names can be 3,000 characters long; you can fill a screen with one of them. With this capability, huge blocks of information can be embedded in Prolog programs.

Some time ago I decided that pure logic programming wouldn't hack it. The system now incorporates explicit iteration constructs to get around the tail recursion limit. Tail recursion was incompatible with the flexible memory organization the design mandated; this recursion requires a specific stack order to avoid dangling references.

A.D.A. Prolog actually was written for a much larger machine. It is a structure-sharing system, which means it wants a large word length, and it is written entirely in C. It cannot be as fast as a system written specifically for the PC, except where the extensions make pure logic programming look ridiculous. But in light of everything else, does it have to be?

Features are constantly being added to A.D.A. Prolog. I invite users to call for an inexpensive update.

Bob Morein
Automata Design Associates
Dresher, PA

DATABASE INTERFACE

The January 1986 article that reviews Q-PRO4 is excellent ("A Data Manager for Intelligent Screen Forms," Chris Christian, p. 126). However, it contains an error that could cause problems for the novice programmer. On page 131, the sentence that says, "A reference to the second element of a state array in file 1 would be **&state[1,2]**" should read "...**&state[1,1]**." A file array is logical and the first element is 0, not 1. Referencing an item in a file in which **item** consists of a single element as **&item[1,1]** will generate an error while **&item[1,0]** will not.

I have been using Q-PRO4 for more than a year and find it to be most powerful in developing a good screen-to-operator interface. The product's documentation, however, is difficult. Mr. Christian's article provides an overview that should much reduce the learning curve for first-time users.

Truman Garinger
Brookings, OR

Array references in the Q-PRO4 language start with element 0, as Mr. Garinger states. The variables in the state array in file 1 are thus referenced by **&state[1,0]**, **&state[1,1]**, **&state[1,2]**, *and so on. One distinction that used to be made between high-level and low-level languages was the former's penchant for relative 1 addressing versus relative 0. This is no longer true, and Q-PRO4 uses the more popular relative 0 address reference.*

—Chris Christian

After two years of subscribing to *PC Tech Journal* and reading untold numbers of hardware and software reviews, I was delighted to finally see a software tool that I use, Q-PRO4 from Q-N-E International, merit its own article.

The only point of contention I have with Mr. Christian is his insistence that Q-PRO4 programs need menus. I have written more than 20 applications programs in one system using only one

Mark Williams

Now the biggest name in C compilers comes in a size everybody can afford.

Let's C.™

$75

Introducing Mark Williams' $75 C compiler. Want to explore C programming for the first time? Or just on your own time? Now you can do it in a big way without spending that way. With Let's C.

This is no little beginner's model. Let's C is a powerful programming tool, packed with all the essentials of the famous Mark Williams C Programming System. The one chosen by Intel, DEC, Wang and thousands of professional programmers. The one that wins the benchmarks and the reviewers' praise:

"(This compiler) has the most professional feel of any package we tested…"—BYTE

"Of all the compilers reviewed, (it) would be my first choice for product development."—David W. Smith, PC WORLD

And now for more big news. Get our revolutionary csd C Source Debugger for just $75, too. You can breeze through debugging at the C source level ignoring clunky assembler code.

Affordable, powerful, debuggable. Mark Williams Let's C is the big name C compiler at a price you can handle. Get your hands on it now.

Mark Williams Let's C

- **For the IBM-PC and MS-DOS**
- **Fast compact code plus register variables**
- **Full Kernighan & Ritchie C and extensions**
- **Full UNIX™ compatibility and complete libraries**
- **Small memory model**
- **Many powerful utilities including linker, assembler, archiver, cc one-step compiling, egrep, pr, tail, wc**
- **MicroEMACS full screen editor with source**
- **Supported by dozens of third party libraries**
- **Upgradeable to C Programming System for large scale applications development**

Let's C Benchmark Done on an IBM-PC/XT, no 8087. Program: Floating Point from BYTE, August, 1983.

	Exec Time in Seconds
Let's C	134.20
MS 3.0	347.45

Use this coupon or charge by calling toll-free: 1-800-MWC-1700. In Ill. call 312-472-6659.

ORDER NOW! 60-DAY MONEY BACK GUARANTEE!

Mark Williams Let's C

Please send me:

_____ copies of Let's C and _____ copies of csd (C Source Debugger) at $75 each. (Ill. residents add 7% sales tax.)

☐ Check ☐ Money Order ☐ Visa, MasterCard or American Express

Name_____________________

Address_____________________

City_____________ State_______ Zip_______

Card #_____________ Exp. Date_______

Signature_____________________ TJ0686

Mark Williams Company
1430 West Wrightwood
Chicago, Illinois 60614

CIRCLE NO. 205 ON READER SERVICE CARD

"choose another screen" menu. Hierarchical menus are the bane of good user interfaces: a transaction must never be spread over two screens if one will do.

Two aspects of Q-PRO4, the source file editor used in the FB utility and the below-average performance of the file structure, were criticized, and rightly so, but they are no longer a problem. A new file structure was debuted in January: a self-balancing B-tree.

In addition, a colleague and I recently brought out a utility through which a user may program in his favorite editor. Called QUTE, it is faster than using FB. This utility can take apart the old FB format into screen fields and ASCII parts, put them back together after editing, find unpaired constructs, and clean up indentation inconsistency.

Other utilities for Q-PRO4 are currently on the market, including XREF and QLINT. Q-N-E International also produces an author's package to encrypt source code, a nice touch.

I look forward to many more years of subscription to your magazine.

Richard Pearlman
Indefinite Energy
New York, NY

With the data manager series, we try to give our readers a feel for the application style that naturally follows from the strengths and weaknesses of each product. It is true that menus are not a requirement of a Q-PRO4 application. Q-PRO4's ability to work with menus quickly and on a field-by-field basis distinguishes it from other products reviewed in the series. We want our readers to understand that in their selection of products for various tasks.

Q-N-E International released Q-PRO4 4.0 in January with the revised program and index organization to which we referred at the conclusion of our article. The new release uses DOS file handles, so that paths and explicit subdirectory references can be used. With these improvements, Q-PRO4 has completed its migration to the PC environment. No changes in source code are required. As Mr. Pearlman points out, various support utilities are available, several from Q-N-E International, and others from user groups.

—Chris Christian

LEGAL WRITES

This is a short fan letter to say how much I have enjoyed Max Stul Oppenheimer's column in *PC Tech Journal.* His February 1986 article ("The Total Solution," Legal Brief, p. 193) certainly hit home. As a computer consultant specializing in turnkey CAD systems using the AutoCAD software package, I frequently face the problem of customers who wish to purchase their systems in a piecemeal fashion, while expecting total support from us as the vendor of the main piece of software used. I am sure you can understand why I find this unfair: it forces me to take up the burden of supporting equipment on which I was not allowed to make a profit. Such a profit is my compensation for that support. As such, I obviously am interested in having my customers purchase as much of their systems from me as possible. A turnkey solution works—a piecemeal one may not.

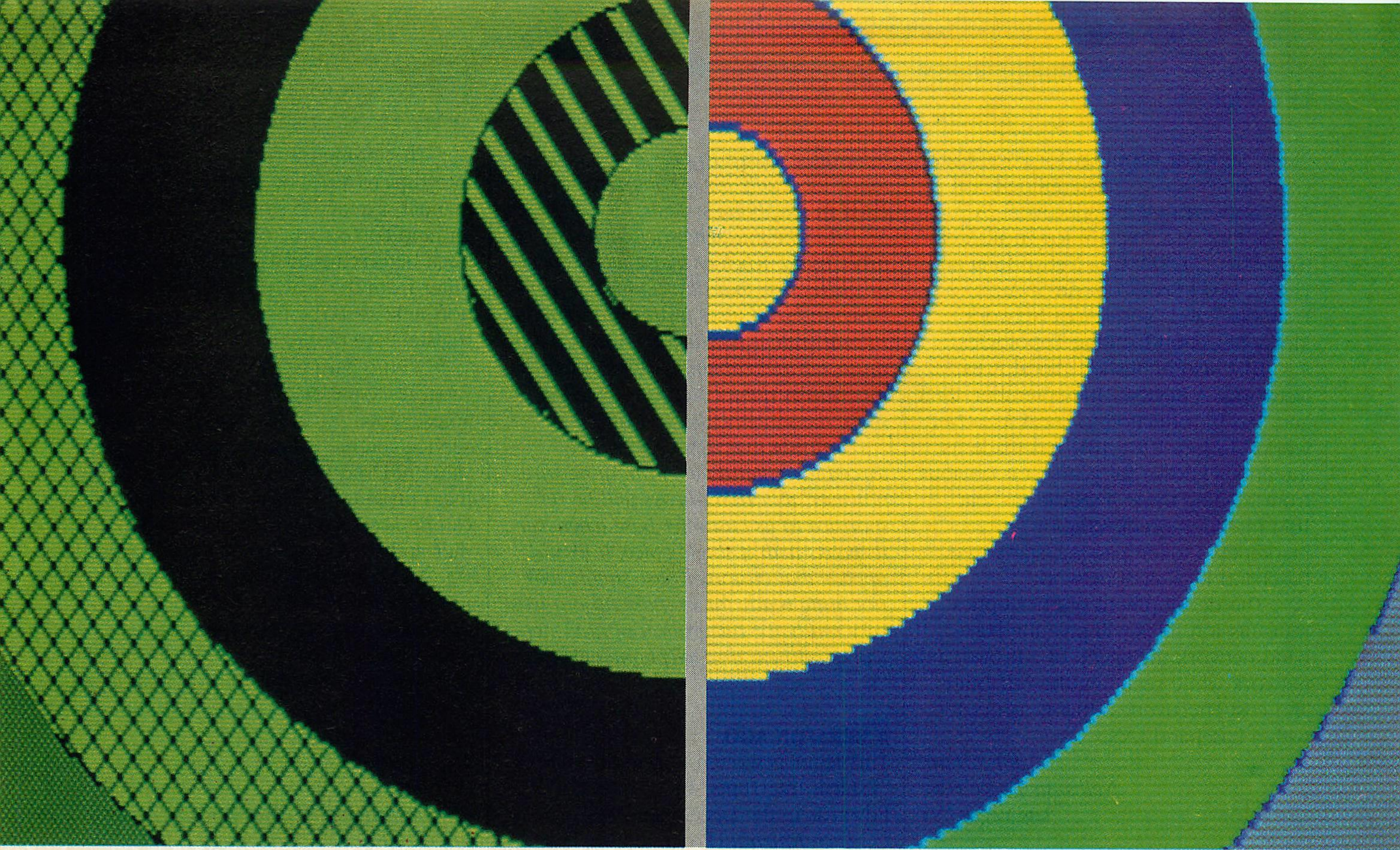
Let me state categorically, however, that I am not in the "you must buy your plotter pens from me" school of VARship. I merely note that when a VAR expresses his fear that an uneducated user accidentally will cripple a system, giving as his reason his concern that the user will not be able to contend with "the complicated interrelationship of hardware and software," he is speaking the absolute truth. What is more, increasing sophistication of present-day users notwithstanding, I do not believe that most of my customers could purchase all the components I have provided off the shelf and configure them into a working system. The qualification, of course, is that I deal in a highly sophisticated and particular piece of software; configuring a Lotus 1-2-3 system is an easy task by comparison.

Once again, let me thank you for producing such an informative and enjoyable column. I am not aware of any other publication that consistently provides a column that is devoted exclusively to the legal issues of the microcomputer world. I find the legal perspective fascinating.

Matt Richard
Premier Design Systems, Inc.
Baltimore, MD

LAN SCOPES

In the January 1986 Directions column, "The LAN of IBM" (p. 9), I got the impression that Will Fastie was intimating that IBM had capitulated to its customers' demands that the RF broadband technology of its PC-Network be abandoned in favor of the baseband communications technology of the IBM Token-Ring Network. A little research into the history of IBM involvement with networks reveals that, in 1983, IBM submitted five papers to the IEEE 802 Committee on local area networks. They were:



Hercules compatible graphics: 720 x 348.

Plantronics compatible graphics: 640 x 200 x 4 colors.

Introducing The Paradise Hi-Res Graphics Card.™ Two Upgrades For The Price Of One.

Now you can have much higher resolution than your current video card gives you. Without buying new software. Or an expensive monitor.

All you need is the new Paradise Hi-Res Graphics Card.

It gives you the highest resolution possible on a standard monitor. With monochrome displays, you get Hercules compatible 720 x 348 graphics. With RGB and composite displays, you get Plantronics compatible graphics: 16 colors in 320 x 200, and 4 colors in 640 x 200. That's two upgrades for the price of one.

And no matter which monitor you choose, the Paradise Hi-Res Graphics Card gives you full compatibility with standard video modes.

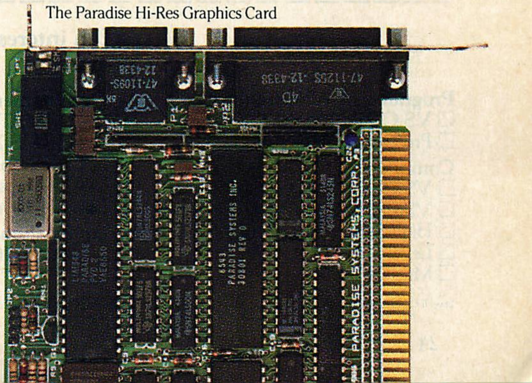
Better yet, you can use your existing software to its best advantage. Because Hercules and Plantronics graphics are widely supported by packages like Lotus 1-2-3, Symphony, Framework, and lots more.

And Paradise gives you all this, plus a parallel port. On one short card.

If you'd like to know more, we'll send you full information. Including a free Hi-Res software compatibility guide. Just call (800) 527-7977, Ext. 252, toll-free. From California, (800) 822-2020, Ext. 252. Or check it out at the best PC dealers.

PARADISE
SYSTEMS, INC

The Paradise Hi-Res Graphics Card



CIRCLE NO. 230 ON READER SERVICE CARD

"Ring Network Topology for Local Data Communications," "Token-Ring Local Area Networks—A Perspective," "A Token-Ring Architecture for Local Area Networks," "Local Area Networking and Higher-Level Protocols: An SNA Example," and "Local Area Network Media Selection for Ring Topologies."

IBM also published "Local Area Networks: A Review" in September 1983 and "An Introduction to Local Area Networks" in July 1984. IBM has made its intentions known for years. It is obvious from reading these publications that IBM intended on developing and marketing a token passing, star-wired, baseband local area network.

The development of fiber optics transmission lines will give the baseband network the speed of a broadband network while retaining the reliability and cost effectiveness that is associated with a baseband LAN.

Cliff K. Fujii
Nellis Air Force Base
Las Vegas, NV

We are familiar with all the papers you quote and with IBM's long-standing research into token-ring technology. However, not all IBM research projects come to commercial fruition, and broadband technology seems to offer many advantages, especially for a company interested in telephone systems.

It is the multichannel capability of broadband, whether wired with copper or glass fiber, that distinguishes it from baseband and gives it an order-of-magnitude greater bandwidth. And it is the multichannel capability of broadband that allows data, voice, and video to share the same physical medium.

I doubt that IBM ever can be forced to "capitulate." I am quite sure, however, that IBM always will be responsive to its customer base whenever it speaks with a single voice. It would appear that customer base simply was not quite ready for broadband but was very ready for integration of data networks into existing telephone wiring.

—WF

A VERSION AGO

This letter is written in reference to "INLINE Interrupts" by Charles C. Edwards in the December 1985 issue of *PC Tech Journal* (Programming Practices, p. 181). While Borland's documentation may have been wrong in older editions, it seems to be correct in the second edition, June 1985 printing of the Turbo 3.0 manual.

The in-line instructions given for interrupt routines on page 214 of the manual save and restore the DS register, and correctly take SP and BP off the stack before the IRET instruction. They differ from the article's recommendation only as to the order in which the registers are placed on the stack.

With regard to accessing global variables, page 215 states: "The way to access global variables in the interrupt service routine is therefore to store the value of Dseq in a typed constant in the main program. This typed constant then can be accessed by the interrupt handler and used to set its DS register."

Mr. Edwards is quite right, though, in advising that you should take anything you read (including this letter) with a grain of salt.

John W. Spalding
Atlanta, GA

Mr. Spalding is correct in pointing out that the Turbo 3.0 documentation concerning interrupt handling is much improved over 2.0, although it is still difficult for a novice to understand.

My article was written when 2.0 was the most current version of Turbo. I am sure Mr. Spalding can appreciate the hours that went into getting this fea-

"Most Significant Product Of 1985– RAMpage!® from AST."

—*PC Week*

LOTUS COMPATIBLE · ASK YOUR DEALER FOR APPROPRIATE CONFIGURATIONS

Lotus®

When we introduced our RAMpage!® solution for IBM® and compatible PCs, we knew we had a winning product. It represents a new plateau in PC enhancement, combining two megabytes of expanded memory with innovative software that lets you immediately use your increased memory power. The resulting solution greatly improves productivity and is adaptable to any business environment.

The most powerful expanded memory product available, RAMpage! and its advanced DESQview™ software allow you to run multiple applications simultaneously, using your existing programs. Including all versions of Lotus 1-2-3,® Symphony,® dBASE™ and Framework™.

So you can sort a data base, write a letter using your favorite word processing application, total a spreadsheet and print a report…all at the same time. No other expanded memory product offers as much.

The people at PC Week, a leading news publication for the PC industry, think our RAMpage! solution is significant. We know you will, too.

To receive our special information package describing AST's RAMpage! solution, call our Customer Information Center today **(714) 863-1333.** Or complete this coupon and send it to AST Research, Inc., 2121 Alton Avenue, Irvine, California 92714.

Yes, I want more information on AST's RAMpage! Solution.
Please send me your special information package today!

Name: ______________________

Title: ______________________

Company: ______________________

Address: ______________________

City: __________ State: ______ Zip: ______

( ) ______________________
Phone:

RAMpage! registered trademark of AST Research, Inc. IBM registered trademark of International Business Machines Corp. DESQview trademark of Quarterdeck Office Systems. Lotus 1-2-3 and Symphony registered trademarks of Lotus Development Corp. The Lotus logo is used by permission of Lotus Development Corp. The Rampage! Expanded Memory Board has successfully passed compatibility testing with 1-2-3 Release 2 and with Symphony 1.1 on an IBM XT with PC-DOS 3.0. Two megabytes of memory were installed. dBASE and Framework trademarks of Ashton-Tate.

CIRCLE NO. 109 ON READER SERVICE CARD

AST RESEARCH INC.

ture to work, considering I was using erroneous documentation.

Also, although the 3.0 manual does mention storing DS in a typed constant for use by an interrupt procedure, it does not show how to fetch this value when the interrupt occurs. Because one of the reasons for working in high-level languages such as Pascal is to avoid the vagaries of machine language, this is an unpardonable oversight. Perhaps this, and the lack of a working example of such a powerful feature, will be rectified in a future version.

The explanation of the use of interrupt procedures and the example put forth will be helpful to readers still using earlier versions of Turbo.

—Charles C. Edwards

CALLing BASIC

Your readers might be interested to learn about an important difference between the Microsoft QuickBASIC compiler and the more expensive IBM BASIC compiler. I use both and was very surprised to find that among the unsupported statements in QuickBASIC were BSAVE, BLOAD, and absolute CALLs. The absolute CALL is of particular concern. Many small machine language routines are BLOADed or POKEd into memory in many BASIC programs, and then CALLed at the address loaded. This is impossible in QuickBASIC, though IBM allows it.

Microsoft's only answer to this problem is that it was impossible to execute an absolute CALL because its compiler uses a relocatable runtime module. If that makes it impossible, then how did IBM do it?

Microsoft claims in the media that except for Indexed File supports and large arrays, its QuickBASIC is completely compatible with the IBM versions. It ain't necessarily so.

Jay Cooprider
President
Software Dynamics Corporation
Oklahoma City, OK

True enough, IBM supports absolute CALLs and Microsoft does not. Neither compiler uses memory the same way as BASICA, of course, so it is a tricky business to translate such CALLs. Both vendors suggest alternate methods: Microsoft suggests assembly language routines be linked in after compilation; IBM advises loading an assembly language routine's hex code into an integer array and executing it from there.

—DB

Choosing a LAN System.

New Report Outlines Steps For LAN Evaluation.

Choosing the best local area network hardware for your particular installation is often a bewildering process. There are many options and few evaluation tools.

A new report developed by Novell, Inc., is designed to offer help. The study, *LAN Evaluation Report 1986*, examines all of the hardware issues that affect LAN performance. It includes an analysis of many LAN products and a series of benchmark tests.

A key element of the study is the addition of an evaluation system. The system provides a mechanism for matching site needs to specific hardware. Whether a new network is being planned or an existing site is being upgraded, the study will be useful in the performance evaluation of any proposed network. As a network operating system developer and system reseller, Novell has experience with LAN products from system support through installation and day-to-day operation. Novell's NetWare® network operating system supports 30 different LAN configurations including the NETBIOS-compatible LANs. Information in the *LAN Evaluation Report 1986* is based on that experience.

LAN Hardware Choices.

LANs are highly modularized architectures. The rich assortment of available LAN components provides flexible building blocks with which to customize networks.

System planning starts with the LAN hardware: the network interface card (NIC) and cabling. Personal computers and other machines are attached to a LAN by plugging the NIC into the PC expansion bus and attaching the LAN cable to the NIC.

"The *LAN Evaluation Report 1986* includes an analysis of many LAN products as well as a series of benchmark tests."

LAN hardware systems analyzed in the study are:

- AT&T StarLAN
- Corvus Omninet
- Davong MultiLink
- Gateway G-Net
- IBM PC Network
- Interactive Systems LAN/PC
- Nestar PLAN 2000
- Novell S-Net
- Proteon ProNET
- Standard Microsystems ARCNET
- 3Com EtherLink
- 3Com EtherLink Plus

The IBM Token Ring Network is also analyzed; however production NICs were not available at the time the report was written and could not be included in the benchmark tests.

The report analyzes each NIC according to its access scheme, raw bit rate, on-board processing, and NIC-to-host transfer method.

NICs divide information into message units called packets, transmit the packets at a certain speed, and manage the transmission and receipt of those packets. In other words, NICs implement a hardware protocol.

One of the points made in the analysis and benchmarks is that the way a protocol is implemented is often more significant to performance than the protocol itself. Many LAN vendors are actively modifying their NIC designs to improve performance without any change in the basic protocol.

Network Servers.

The network server manages all network requests and data storage functions. Because of this, the server plays a key role in LAN performance.

Servers come in many different configurations and designs. Some are proprietary boxes that were

specifically designed as servers. Others are personal computers that are functioning as servers.

The *LAN Evaluation Report 1986* analyzes and tests the following servers:

- IBM PC XT
- IBM PC AT
- Novell S-Net
- 3Com 3Server
- Novell 286A and 286B

Processor type is the most obvious difference among these machines. They use the Intel 8088, Intel 80186, Intel 80286 or Motorola MC68000. But other factors are also important in determining server performance, including processor clock cycle speed, wait states, server memory cycle speed, memory channel, and transfer bus channel. All of these factors determine the speed at which data is moved and processed.

One of the jobs that a server handles is sending data to and from the hard disk. The speed of the server cannot alter the speed of the disk channel. If a disk channel can handle reads at 160 kilobytes per second, a faster processor isn't going to change that figure.

A faster server, however, can change the percentage of processor utilization for specific servers. High-performance servers use less of the processor's time for specific operations, freeing the processor to perform other tasks. The result is increased performance.

NetWare Evaluation System.

The *LAN Evaluation Report 1986* contains the NetWare Evaluation System. While benchmark measurements are part of the system, they are designed to be used only as input in the evaluation formula.

The first of the two benchmarks is a measurement of maximum throughput for a LAN/server combination with a single workstation.

The second shows the maximum working bandwidth for a fully loaded network. In this test, six IBM PC AT workstations were attached to the various LAN/server combinations. The network was driven to its maximum traffic capability. Throughput results from all stations were totaled to show the maximum bandwidth of each network.

"The NetWare Evaluation System provides an excellent method of LAN performance comparison."

A formula for workstation usage is then developed based on specific values for a particular site. Five categories of network users are defined and used in establishing this site profile.

As a final step, the benchmark results and site profile are applied to a formula which shows the throughput of the proposed system.

Answers generated by the formula provide several useful pieces of information. The results from the formula should be approximately the same as the maximum throughput in a single station test.

If the working bandwidth of the proposed system is much greater than the single station throughput, this indicates that the LAN/server combination has more power than can be utilized by the proposed network. If the working bandwidth is much lower than the single station throughput, it indicates that the LAN/server combination will be overburdened.

The evaluation system also provides a method of performance comparison. Desired performance is usually expressed in terms of floppy or hard disk speed. The study contains data on the standalone workstation performance of the IBM PC AT and XT with both floppy and hard disks. Using these figures, the formula's results can be compared to desired throughput.

Read the Full Report.

The *LAN Evaluation Report 1986* is available free of charge from Novell. To obtain a copy, call or write Novell Corporate Communications, 748 North 1340 West, Orem, Utah 84057, (801) 226-8202.

NOVELL

MICROSOFT LANGUAGES NEWSLETTER Vol. 1, No. 6

News about the Microsoft Language Family

Your Microsoft® FORTRAN Programs Can Call Microsoft C Library Routines

Microsoft FORTRAN's long-established history includes powerful scientific subroutines drawn from a vast user community. Microsoft C has a rich operating systems background, strong string and bit manipulation support, and growing strength in the program portability arena. The following demonstrates how easily one can call C functions from a FORTRAN program.

Spawnlp creates and executes a child process. In this example, we suspend the parent program while the child program executes. When the child program terminates, the parent program resumes execution.

Spawnlp is declared in C as follows:

```
int spawnlp (mode, path, arg0, arg1, ..., argn)
int mode; char *path, *arg0, ..., *argn;
```

We declare the interface to FORTRAN with this program fragment:

```
integer*2 spawn
interface to integer*2 function spawn [c, varying, alias:'spawnlp'] (mode)
integer*2 mode
end
```

Spawn is the function name we will use from FORTRAN. We declare the return type of spawn to be integer*2. [c] indicates the C language. [varying] tells that a variable number of arguments may be passed. An [alias] is used because the C name for the function spawnlp has 7 characters; names in FORTRAN are only significant to 6 characters. The string arguments are undeclared in the interface and assumed to be passed from FORTRAN by value.

The function can now be invoked as follows:

```
i = spawn(0,loc('exemod'c), loc('exemod'c), loc('demoexec.exe'c), int4(0))
```

The C spawnlp function expects addresses of strings, not actual characters, so we use the LOC( ) function. C strings differ from normal FORTRAN strings; we specify these by the "c" after each closing quote. We use INT4(0) to pass the last parameter, a C NULL pointer (32-bit integer zero).

Starting with the release of Microsoft FORTRAN 3.3, Pascal 3.3, Macro Assembler 4.0, and C Compiler 3.0, these Microsoft languages are designed so libraries and subprograms written in any one can be used in any other. Any C routine (not just spawn) can be interfaced to FORTRAN. And they are supported under both MS-DOS® and XENIX,® for additional program portability.

For more information on the products and features discussed in the Newsletter,
write to: Microsoft Languages Newsletter
16011 NE 36th Way, Box 97017, Redmond, WA 98073-9717
Or phone:
(800) 426-9400. In Washington State and Alaska, call (206) 882-8088. In Canada, call (416) 673-7638.

Latest DOS Versions:

Microsoft C Compiler	3.00
Microsoft COBOL	2.10
Microsoft FORTRAN	3.31
Microsoft Macro Assembler	4.00
Microsoft Pascal	3.31
Microsoft QuickBASIC	1.02

Look for the Microsoft Languages Newsletter every month in this publication.

The Portable II

Compaq has taken the boredom out of IBM compatibility with its smaller, lighter AT-in-a-suitcase.

The IBM PC standard for microcomputers has done more for the acceptance of personal computers on the desktop than anything since VisiCalc. For watchers of the computer industry, however, it has created a horrible side-effect: boredom.

Virtually every manufacturer of personal computer systems has acknowledged that adherence to the IBM standard is absolutely essential. And that spells boredom. The desktop machine becomes a commodity product and purchase decisions can be safely made solely on the basis of price. The flavor of the IBM machine is vanilla and other machines are just different brands of vanilla. About the only way to make vanilla more interesting is to add real vanilla beans to the mix and charge a premium price.

Compaq Computer Corporation brings excitement to vanilla. Its legendary success needs no repeating here. The elements of that success are a PC in a suitcase, a better PC for the desktop, a better AT for the desktop, *and* an AT in a suitcase; these machines are Compaq's vanilla beans.

Compaq has now added another bean. In the middle of the rise of the laptop, Compaq has (perhaps only for the moment) eschewed the tiny machines in favor of a better version of its original concept. For the excellence in both its design and conceptualization, *PC Tech Journal* names Compaq's Portable II (Model 3) the June 1986 Product of the Month.

If ever there was a PC to fall in love with, this is it. The Portable II delivers a full-function AT compatible in a suitcase that is 30-percent smaller (to fit under an airline seat) and 17-percent lighter (just over 26 pounds) than the other Compaq portables on the market. Its 80286 processor operates at 8 MHz but can be adjusted (from the keyboard and while running) to operate at the standard AT's slower 6-MHz rate should that be necessary for software compatibility. The Portable II comes with 640KB of memory; one 360KB, third-height diskette drive; one 10MB, 3½-inch, third-height hard disk; a serial and parallel port; a realtime clock; and two expansion slots. A memory expansion option increases memory to 2.1MB without using a slot. Another Compaq option takes up one slot and expands memory to 4.1MB.

The Portable II includes the nine-inch, dual-mode display of its larger siblings that delivers high-resolution text and IBM Color Graphics Adapter (CGA) compatibility. In graphics mode, colors are represented by shades of green on the monochromatic display.

This recitation of facts and features hardly provides a basis for love, however. There is a subjective element to the Portable II, a feeling that comes on as the time spent using the machine rises. This is a responsive, peppy, agile machine. It will do a lot of work quickly. There is no feeling of being on a non-IBM machine; except for the logo, this comes as close to matching an AT as is legally possible.

The Portable II is also beautiful. It is possessed of clean lines, attractive styling, and an extra touch of color, subdued though it is. It looks sharp, but not threatening. It has been *designed.*

Perfection in computers is an elusive goal; a number of criticisms might be leveled at the Portable II. The most immediate is likely to be Compaq's choice of a 10MB disk instead of the more standard 20MB. Just after IBM's price cuts, Compaq announced its 20MB Model 4 priced at $4,999. *PC Tech Journal* was unable to obtain a unit for examination; if the disk performs like the Model 3 drive, the Model 4 is the more desirable version.

At $4,799 the Portable II could be considered a bit expensive—especially in light of only 10MB of disk and a CGA display. Compaq responds that it is selling all the machines it can build—a point hard to argue.

Another, milder complaint concerns Compaq's choice of the CGA instead of the Enhanced Graphics Adapter (EGA), which seems to be an emerging standard. With large-scale chips now readily available, EGA capability might have been a better idea. However, actual work on the machine leaves the user with the impression of more than satisfactory graphics for everyday use. Even a product like Microsoft Windows, with all of its overhead, runs superbly on the Portable II simply because the number of bits that need to be manipulated for graphics on a CGA is one-seventh that of the EGA. The 8-MHz processor speeds up those fewer manipulations and gives Windows the appearance of achieving far better performance than a standard AT can deliver.

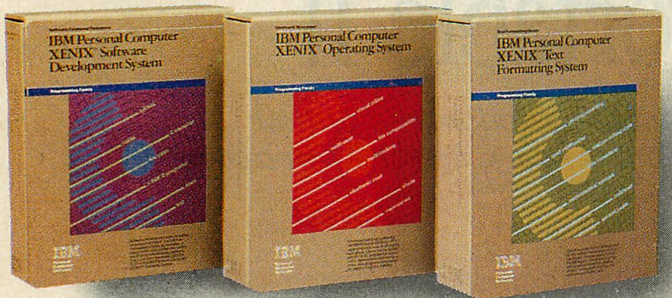
One compromise forced on the Portable II by its smaller size is the keyboard. The function keys have been moved to the top, but the keyboard duplicates all the problems inherent in the AT keyboard. Coincidentally, IBM has introduced a new keyboard for the RT, AT, and XT, which solves some problems and creates new ones.

None of these criticisms adds up to a major complaint. The Portable II has too many features to like. The machine is destined to be quite popular on the desktop, just as its siblings have been, but because it is smaller, faster, and lighter than other luggables, the Compaq Portable II will also be a favorite for software demonstrations and the delivery of full desktop power to out-of-office situations.

This machine is a winner.

Compaq Portable II, Model 3: $4,799
Compaq Computer Corporation
20555 FM 149
Houston, TX 77070
713/370-0670
CIRCLE 353 ON READER SERVICE CARD

Hardware, software, and other developments for the IBM PC family



IBM's XENIX version 2.0

New model IBM PC/AT

FROM IBM

IBM Corporation has announced the **PC Convertible**, the newest and smallest member of the IBM family. This full-function PC, which weighs less than 13 pounds, features an 80C88 microprocessor and 256KB of user memory. The Convertible is equipped with an 80-column-by-25-line detachable liquid crystal display, a built-in 78-key keyboard with full-size typing keys, and dual 3½-inch diskette drives. Each diskette drive has a 720KB capacity, equal to more than 350 typewritten pages. The Convertible includes a battery pack and an AC adapter that can operate the system while simultaneously recharging the battery pack. $1,995.

A new model **PC/AT**, also announced by **IBM**, features an 8-MHz microprocessor that is 30 percent faster than previous AT microprocessors. This latest AT is equipped with a 1.2MB diskette drive, a 30MB fixed-disk drive, a serial/parallel adapter card, and 512KB of user memory, which is expandable to 10.5MB with new memory expansion options. The new machine comes with IBM's 101-key Enhanced PC Keyboard, which separates the typing area, the calculator-style keypad, and the cursor and screen control keys. This AT is compatible with the PC 3½-inch external diskette drive. $5,295.
IBM Corporation, Entry Systems Division, P.O. Box 1328, Boca Raton, FL 33429-1328; Contact the local IBM dealer, 800/426-2468
CIRCLE 337 ON READER SERVICE CARD

The first issue of the ***IBM Technical Directory*** has been released. This pamphlet lists all technical books and reference materials produced by IBM for its family of Personal Computers and software products. The books and materials listed are available through authorized IBM PC dealers as well as by calling the toll-free number provided here and in the pamphlet. Free of charge.
IBM Technical Directory, P.O. Box 2009, Racine, WI 53404; 800/IBM-PCTB, or contact the local IBM dealer, 800/426-2468
CIRCLE 301 ON READER SERVICE CARD

The **IBM 3295 Display Adapter** attaches the 3295 Plasma Monitor to the PC, PC/XT, and PC/AT. (The Plasma Monitor is a large-screen, high-resolution monitor capable of 160 columns by 64 rows of alphanumeric or 960 columns by 768 rows of all-points-addressable graphics. The monitor has a thin profile and a small footprint.) The Display Adapter provides applications programs with two interfaces: first, a monochrome display interface to provide an 80-by-25 alphanumeric window compatible with existing programs written for the monochrome display; second, a low-level program interface that allows new or enhanced applications to use its full-screen capabilities. $1,495.

Also from IBM comes an advanced-function, short-card version of its **3278/79 emulation adapter** for the PC. The adapter enables the PC to emulate an IBM 3278 monochrome display station or an IBM 3279 color display station. Users can transfer files to and from host systems in 3270 data stream environments. Advanced capabilities provided by the new adapter include multiple concurrent 3270 sessions and new connectivity options. $595.

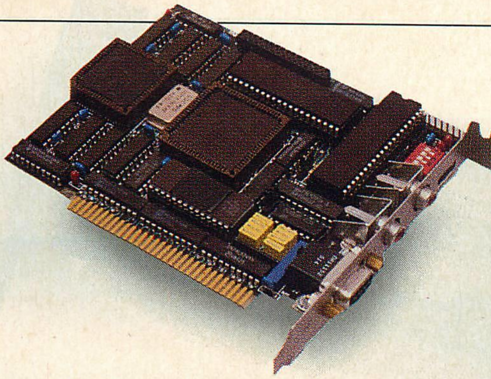
Several new products that enhance the capabilities of the **IBM Token-Ring Network** have been announced. The **Token-Ring Network/PC Network Interconnect Program** provides a connection between the IBM Token-Ring Network and the IBM PC Network. A dedicated PC running only the interconnect program is attached to both using adapters. This permits devices on one network to communicate with devices on the other. $495.

The **Token-Ring Network Basic Input/Output System (NETBIOS) Program** provides a NETBIOS programming interface for the Token-Ring Network. The program allows applications programs to be written for operation on both the Token-Ring Network and the IBM PC Network. $35.

With the **IBM Asynchronous Communications Server Program**, PCs on the Token-Ring Network or the IBM PC Network can access ASCII applications via switched communication lines. Attached PCs can share this communication server and its asynchronous communications lines. $495.

Also available for use with the Token-Ring Network is the **Advanced Program-to-Program Communication for the IBM PC (APPC/PC)**. This program supports the SNA applications programming interface (LU 6.2 and PU 2.1) and allows program-to-program communication over the IBM Token-Ring Network as well as synchronous data link control (SDLC) communication links. $150.
IBM Corporation, Information Systems Group, Rye Brook, NY 10573; Contact the local IBM dealer, 800/426-2468
CIRCLE 302 ON READER SERVICE CARD

Version 2.0 of **IBM's XENIX Operating System** has been announced. This version offers many functional enhancements but maintains binary compatibility with programs developed under 1.0. Version 2.0 supports up to 3MB of memory, provides dynamic memory management and protection, includes a hierarchical file system and three command interfaces, and supports the 80287 coprocessor. The **IBM PC XENIX Operating System Extensions** provide additional end-user functions in the XENIX environment that enhance compatibility with other IBM UNIX System V derivative products. The enhancements include a full-screen editor with windowing, a function-key driven user inter-



SigmaEGA! by Sigma Designs

IBM PC Convertible

face, UNIX System V accounting functions, and versatile intersystem communication facilities. XENIX 2.0, $429; upgrade, $299; XENIX Extensions, $475; upgrade, $235.

Also introduced is **version 2.0** of IBM's **XENIX Software Development System**, which now includes a C language compiler that supports arrays larger than 64KB and an improved assembler with macro support and several new library routines and system calls. $499; upgrade, $299.

Version 2.0 of IBM's **XENIX Text Formatting System** contains extensive text processing components that simplify the production of technical reports, memoranda, formal papers, and documentation. $159; upgrade, $99.

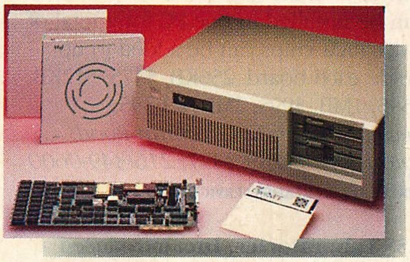
IBM Corporation, Boca Raton, FL 33429-1328; Contact the local IBM dealer, 800/426-2468

CIRCLE 303 ON READER SERVICE CARD

HARDWARE

Intel Corporation has introduced a group of products that can link host computer systems into LANs in the microprocessor development laboratory. The Open Development Networking products include **OpenNET Network Resource Manager (NRM), VAX Link R2.1**, and **OpenNET PC Link and Compilengine**. This network allows dissimilar computers and operating systems to be joined in a single network.

The NRM is a high-capacity file server (up to 560MB) that allows any workstation on the network to gain transparent access to share files stored in the NRM's protected hierarchical file system. VAX Link R2.1 supports file copy and distributed job control between VAX/VMS R4.2 systems and the OpenNET NRM. OpenNET PC Link uses an add-in EtherNet controller and OpenNET software to allow PC/XTs, PC/ATs, and compatibles to access files on an OpenNET server as if the files were on the PC itself. Compilengine is a specialized server that imports time-consuming compilations and link/locates from systems on the network through distributed job control. Compilengine features



Intel's OpenNET

an 8-MHz 80286 CPU, a 1MB zero-wait-state RAM, and a 40MB Winchester hard disk with a controller that has an 80186 and 32KB of track caching. OpenNET NRM with 40MB hard disk, $14,995; with 140MB hard disk and 60MB streaming tape drive, $23,995; OpenNET PC Link, $1,250; Compilengine, $13,995; VAX Link, $7,500.

Intel Corporation, Literature Dept. W274, 3965 Bowers Avenue, Santa Clara, CA 95051; 800/548-4725

CIRCLE 304 ON READER SERVICE CARD

The **PC Supercharger** from **Dynatec Systems, Inc.** is a speed optimizer that enables a PC, PC/XT, or compatible to run at twice its normal operating speed. Because this small plug-in board uses an 8088-1 microprocessor, it is 100-percent compatible with a system's existing hardware and software. The Supercharger is totally transparent to the system and the user. It plugs directly into the 8088 socket, with no modification to the system required. All expansion slots are free for memory, video, and other peripheral boards. $279.95.

Dynatec Systems, Inc., 870 E. 9400 South, Suite 103-B, Salt Lake City, UT 84070; 801/572-6867

CIRCLE 310 ON READER SERVICE CARD

The **SigmaEGA!** is a high-resolution graphics board from **Sigma Designs** that incorporates graphics standards for the IBM Enhanced Graphics Adapter, the IBM Color Graphics Adapter, the IBM Monochrome Display Adapter, and the Hercules Graphics Adapter. The SigmaEGA! includes 256KB of on-board standard memory, which enables users to run all EGA graphics modes without purchasing supplementary memory expansion modules. $595.

Sigma Designs, 2023 O'Toole Avenue, San Jose, CA 95131; 408/943-9480

CIRCLE 305 ON READER SERVICE CARD

A keyboard designed to work with micro-to-mainframe 3270 emulation packages has been developed by **Key Tronic Corporation**. The **KB 3270/PC Keyboard** has a 122-key layout and is plug-compatible with the PC, PC/XT, and PC/AT. The KB 3270/PC works with many 3270 emulation packages, including products from CXI, FORTE, Attachmate, and IRMA. $326.

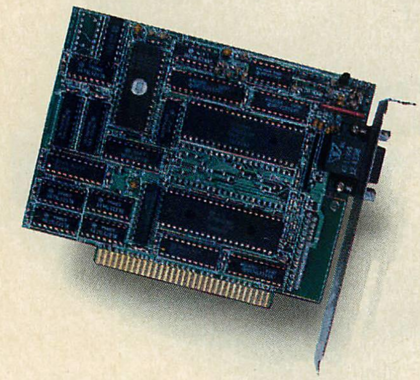
Key Tronic Corporation, P.O. Box 14687, Spokane, WA 99214-0687; 509/ 928-8000

CIRCLE 313 ON READER SERVICE CARD

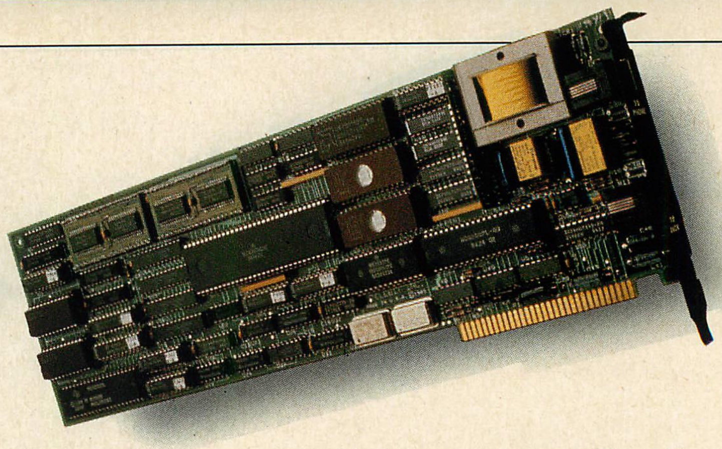
A ring concept LAN is available from **Racore Computer Products, Inc. LANpac** transfers data at the rate of 2 megabits per second. It consists of proprietary LAN ring architecture on a plug-in network interface card and the necessary cable and connectors. LANpac can link 250 terminals in any configuration, with up to 1,000 feet between the nodes. LANpac offers repeaters that can increase the distance between nodes up to 10,000 feet. It runs with IBM PC network software on the PC, PC/XT, PC/AT, PC*jr*, and compatibles. $295 per node.

Racore Computer Products, Inc., 10 Victor Square, Scotts Valley, CA 95066; 800/325-1833; in California, 800/255-7227

CIRCLE 314 ON READER SERVICE CARD



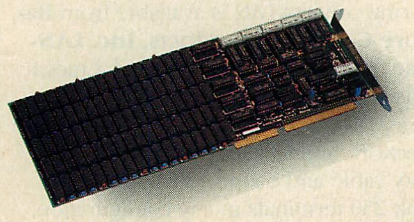
Paradise Systems' Short Color Card



IRMA's FASTLINK internal version

The **Short Color Card** and the **Short Mono Card**, both from **Paradise Systems, Inc.**, are 100-percent compatible, short-slot replacements for the IBM monochrome and color graphics display adapters. The Short Color Card provides flicker-free scrolling, eliminating snow, and displays 25 lines of 40 or 80 columns. Its character box is 8 by 8 pixels, and it offers a text resolution of 320 by 200 or 640 by 200. It includes graphic modes of 320 by 200 with four colors and 640 by 200 with two colors. The Short Mono Card offers 25 lines by 80 characters, has an 8-by-14-pixel character box, and provides text resolution of 640 by 350. The Short Mono Card includes a parallel port. Color Card, $179; Mono Card, $199.
Paradise Systems, Inc., 217 E. Grand Avenue, San Francisco, CA 94080; 415/588-6000
CIRCLE 307 ON READER SERVICE CARD

Cheetah International, Inc. has announced a no-wait state, add-on memory board for the PC/AT. The **CHEETAH CARD** adds 2.5MB of memory per card and enables the AT to run up to 30 percent faster. Extra 2.5MB CHEETAH CARDs can be added to a system with the help of Cheetah's C-DISK software: users can configure a 10MB-plus VDISK in memory that runs in a no-wait state. $945.
Cheetah International, Inc., 107 Community Blvd., Suite 5, Longview, TX 75602; 800/243-3824; in Texas, 214/ 757-3001
CIRCLE 306 ON READER SERVICE CARD



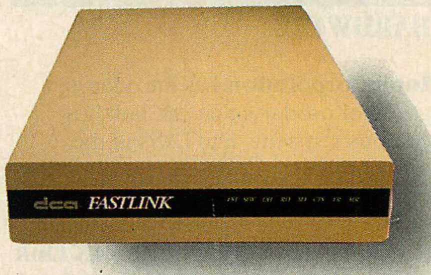
CHEETAH CARD by Cheetah International

640 Plus from **Tecmar, Inc.** is a Lotus-certified memory board that uses a bank-switching technique to give the PC, PC/XT, and PC/AT the power to access memory not normally available under DOS. Each 640 Plus board adds from 256KB to 2MB; as many as four boards can be installed in one PC for a total of 8MB. RAM chips can be added to upgrade each board. 256KB, $475; 512KB, $595; 1MB, $995; 2MB, $,1,395.
Tecmar, Inc., 6225 Cochran Road, Solon, OH 44139-3377; 216/349-0600
CIRCLE 312 ON READER SERVICE CARD

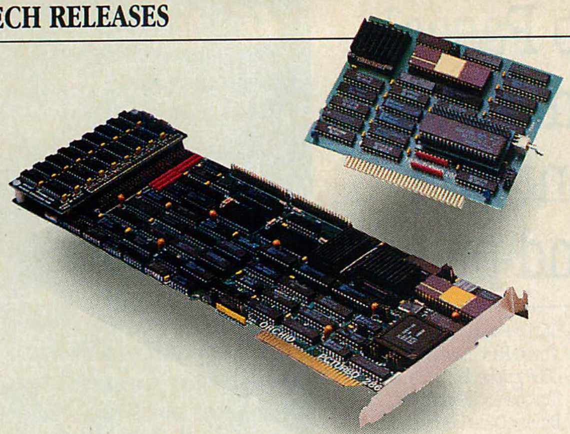
CSS Laboratories, Inc. introduced a motherboard that upgrades the PC/XT, equipping it with the power of a PC/AT. The **XT-286** board elevates the memory, clock speed, and all other functions of the XT to AT performance standards. It incorporates a high-performance, high-speed 16-bit 80286 microprocessor into the system and provides 512KB of on-board memory that can be increased to 4MB with a memory expansion card. Standard clock speed is switch selectable up to 8 MHz. No special software is required. Additional features of the XT-286 include 64KB ROM, which contains Award Software BIOS, 7-channel direct memory access, 16-level interrupt, and a realtime clock. $995.
CSS Laboratories, Inc., 2134 S. Ritchey Street, Santa Ana, CA 92701; 714/540-4141
CIRCLE 317 ON READER SERVICE CARD

MA Systems has announced two new products. The **PC-AT OPTIMIZER** board incorporates the control logic process functions and can be equipped with up to 2.5MB of RAM. A plug-in to the main board adds another 2.5MB of RAM; a second plug-in adds one parallel and two serial ports. A second OPTIMIZER can be added to boost memory to 10MB; a third board can boost it to 15MB. The **PERFORMER-286** board combines the Intel 80286 included in the PC/AT with a VLSI chip that emulates the PC's 8088 processor. It carries 640KB of on-board, 16-bit, high-speed RAM, which increases a system's speed up to 500 percent. The PERFORMER-286 is 100-percent compatible with PC and PC/XT software. PC-AT OPTIMIZER with 5MB plus I/O expander, $2,200; PERFORMER-286, $1,149.
MA Systems, 2015 O'Toole Avenue, San Jose, CA 95131; 800/543-6546; in California, 800/223-3276
CIRCLE 318 ON READER SERVICE CARD

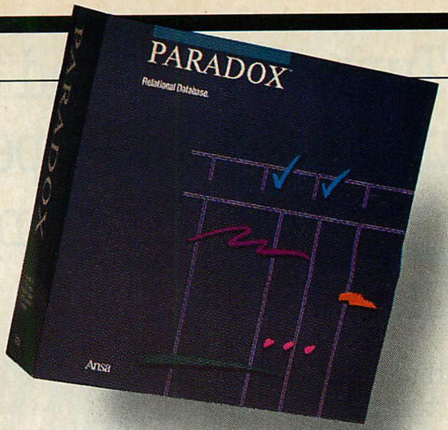
Digital Communications Associates, Inc. (DCA) has announced a new release of **IRMA'S FASTLINK**, a modem that enables users to send information to other PCs, mainframe computers, and minicomputers at speeds up to 18,000 bps over ordinary dial-up telephones without compressing the data. The modem's high rate of speed is due to FASTLINK's technology, developed by **Telebit Corporation**, which combines a multicarrier modulation scheme, digital signal processing, and packet technologies to provide a dramatic increase in the rate of transmission over the public-switched telephone network. Data are transmitted asynchronously with error-detection and correction capabilities. Card version, $1,995; stand-alone, $2,395; upgrades, $99; factory upgrades by DCA, $250.
Digital Communications Associates, Inc., 1000 Alderman Drive, Alpharetta, GA 30201; 404/442-4000
CIRCLE 309 ON READER SERVICE CARD



IRMA's FASTLINK stand-alone version



Orchid's PCturbo 286e (left) and TinyTurbo 286



Ansa's Paradox version 1.1

Two new 80286-based accelerator boards have been introduced by **Orchid Technology**. The no-wait-state **PCturbo 286e** uses an 8-MHz 80286 CPU, an 80287 math processor socket, and a 16-bit internal system bus to improve overall system throughput up to 3.5 times in all applications classes. With the connection of an optional RAM daughtercard, the PCturbo 286e can accommodate the Lotus/Intel/Microsoft expanded memory specification (EMS). The system includes Orchid's productivity software, which can improve throughput 10 to 20 times. $1,195.

The **TinyTurbo 286** is a half-slot accelerator board that achieves processing speeds three times faster than those of the PC or PC/XT. The TinyTurbo 286 uses an 80826 CPU and plugs directly into a host computer's 8088 processor socket. Because the 80286 processor of the TinyTurbo, which is equipped with 16-bit on-board bus and an 8KB static RAM cache, operates independently of, but in tandem with, the PC, it can access all system memory, including EMS and RAM disk. $695.

Orchid Technology, 47790 Westinghouse Drive, Fremont, CA 94539; 415/490-8586

CIRCLE 308 ON READER SERVICE CARD

A hardware/software communications package designed to facilitate communications between LAN systems and mainframe computers has been introduced by **Novell, Inc.** The **NetWare/SNA Gateway** makes it possible for all workstations on a LAN to communicate in a cost-effective way with a host computer. It allows one modem to serve as many as 32 users on the network. The NetWare SNA/Gateway is a nondedicated machine attached to the LAN; it must be a PC/AT or compatible. This computer maintains communications between the host and the network. Features include multiple host sessions, multiple gateways, emulation features, and a hot-key capability. The package includes software, the SNA communications card, an adapter cable, a manual, and eight keyboard overlays. For 1 to 8 sessions, $5,530; 1 to 16 sessions, $6,095; 1 to 32 sessions, $7,495.

Novell, Inc., 1170 N. Industrial Park Drive, Orem, UT 84057; 801/226-8202

CIRCLE 319 ON READER SERVICE CARD

Curtis, Inc. has announced a new version of the **ROMDISK PC EPROM** disk and disk drive emulator accessory card. The **PC RO**, a read-only version of the ROMDISK PC, is available in both a single-sided double diskette (SSDD) equivalent (180KB) and a double-sided double-density diskette (DSDD) equivalent (360KB). Another version of the CPC RO is allows multiple ROMDISK PC ROs or a PC-1 or PC-2 and one or more CPC ROs to operate in a single computer. The user programs the EPROMS on a ROMDISK PC-1 or PC-2 by copying a master disk to ROMDISK PC then transferring the EPROMs to the ROMDISK PC RO. After they have been programmed, multiple copies of the EPROMs can be made on a standard gang EPROM programmer. ROMDISK PC RO SSDD, $395; DSDD, $495; ROMDISK PC-1, $495; ROMDISK PC-2, $595.

Curtis, Inc., 22 Red Fox Road, St. Paul, MN 55110; 612/484-5064

CIRCLE 315 ON READER SERVICE CARD

The **SpeedPac 286**, powered by an Intel 8-MHz 80286 and coupled with 8KB of on-board memory, increases the performance of a PC or PC/XT by as much as 700 percent. This turbo board, introduced by **VICTOR Technologies, Inc.**, fits in a half-size expansion slot and requires no special software. The SpeedPac 286 is backed by a 60-day unconditional money-back guarantee and a full one-year warranty. $595.

VICTOR Technologies, Inc., 380 El Pueblo Road, Scotts Valley, CA 95066-4269; 408/438-6680

CIRCLE 316 ON READER SERVICE CARD

SOFTWARE

Version 1.1 of **Paradox** has been released by **Ansa Software**. The new version of this relational database incorporates a multiple-table update facility to add or revise data in two or more files at once from a single form, a full-featured applications generator to create sophisticated database programs without programming, and new PAL (Paradox Applications Language) commands that allow procedures to be stored in binary form, which results in much faster program loading and operation. Other enhancements include improved memory use for increased speed and performance, an improved procedure for importing ASCII files, automatic updating of reports and forms, and access to DOS from Paradox or PAL programs. Paradox 1.1 is not copy protected. $695; updates are free to registered users, $19.95 to nonregistered users.

Ansa Software, 1301 Shoreway Road, Belmont, CA 94002; 800/547-3000; in California, 415/595-4469

CIRCLE 320 ON READER SERVICE CARD

Arrix Logic Systems, Inc. has announced **APS/microDCF**, a modular text processing system compatible with IBM's mainframe Document Composition Facility (DCF). APS/microDCF consists of three program modules: a text editor (APS/SPF), a high-performance text formatter, and a printer manager. Functional capabilities include multi-column formats, an automatic table of contents, a list of illustrations, a back-of-book index, and referencing of headings, footnotes, and figures. User-written macros and logical symbols are supported, and text and graphics can be mixed on a page. $695.

Arrix Logic Systems, Inc., 2465 E. Bayshore Road, Suite 301, Palo Alto, CA 94303; 800/268-3599

CIRCLE 332 ON READER SERVICE CARD

Reflex, The Analyst upgrades and adds the new Reflex Workshop!

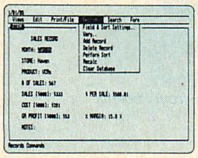
Why running your business without Borland's Reflex and the new Reflex Workshop is an act of blind faith

Running a successful business isn't something you can do with your eyes shut, but no matter what business you're in, Reflex™ and the new Reflex Workshop™ give you all the tools and views to see what all the numbers look like.

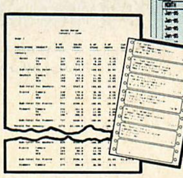
Using Lotus 1-2-3® or dBASE® without Reflex is like driving at night without lights

Products such as 1-2-3 or dBASE can do the numbers for you, but you may still not get the picture—simply because they can't show you analytical graphs and pictures of your data, nor can they analyze and summarize all the information you manipulate like Reflex can.

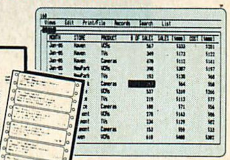
Reflex gives you five graphic ways of looking at your data, five different ways of analyzing your information.



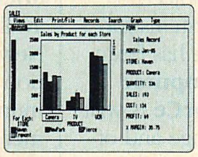
The **FORM VIEW** lets you build and examine your database.



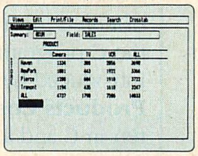
The **REPORT VIEW** allows you to generate everything from mailing labels to sophisticated reports. You can use database files created with Reflex or transferred from 1-2-3, dBASE, PFS: FILE, and other applications.



The **LIST VIEW** lets you put data in tabular list form just like a spreadsheet.



The **GRAPH VIEW** gives you instant interactive graphic representations.



The **CROSSTAB VIEW** gives you amazing "cross-referenced" pictures of the links and relationships hidden in your data.

The best just got better. Introducing Reflex 1.1 NEW!

The new Reflex 1.1 with extended memory support allows you to manage huge databases of up to 8 megabytes of RAM, 32,000 records, and 250 fields per record with the now-legendary "Reflex Lightning Speed."

Furthermore, Reflex 1.1 with its EGA support displays 40 lines of information in its spreadsheet-style List View, compared to less than 25 lines displayed by traditional spreadsheets.

SPECIAL OFFER!

If you already bought Reflex 1.0, get Reflex 1.1 and the Reflex Workshop for only $59.95

Because you bought Reflex from us, you're "our kind of people." And since we're not the "take-the-money-and-run" kind of company, you can upgrade to Reflex 1.1 and the Reflex Workshop for only $59.95. If you prefer to simply upgrade to Reflex 1.1, you can do that for only $10.

SPECIAL OFFER!

*You get Reflex 1.1 and the Reflex Workshop for only $199.95**

Sold separately, the new Reflex Workshop is $69.95 and Reflex is $149.95, totaling $219.90—but you can get them both for a limited time only, at an amazing $199.95. So act now, rush to your nearest dealer, call us, or clip the coupon and put Reflex 1.1 and the Reflex Workshop to work for you right away!

NEW! Introducing the Reflex Workshop
Only $69.95

A major addition to Reflex, the new Reflex Workshop gives you a wide range of analytical tools written for specific applications. You can use these tools "as is" or modify them to suit your analytical and business needs. What you have to work with right away are 25 different tools:

For Finance/Accounting:
- Business Expense Tracking
- Petty Cash Tracking
- Line of Credit Tracking & Analysis
- Accounts Receivable Tracking & Aging Analysis
- Purchase Order Entry & Analysis
- Purchase Order Tracking System
- Leasing Inventory Management
- Asset Inventory Tracking
- Cash Management Trial Balance

For Administration:
- Membership Dues Tracking and Analysis
- Mail List
- Time Management
- Appointment Scheduling
- Applicant Tracking & Inquiry System
- Facilities Planning

For Sales & Marketing:
- Sales Lead Tracking & Analysis
- Store Check Inventory Analysis
- Sales Analysis
- Trend Analysis
- Research Questionnaire Analysis

For Production and Operations:
- Manufacturing Quality Assurance Tracking
- Assembly Repair Turnaround Tracking
- Commercial Real Estate Tracking & Analysis
- Project Scheduling
- Product Cost Analysis and Control

"The best database around . . . at any price.
Jean Lockwood, Computer Retail News

Everyone agrees that Reflex is the best-looking database they've ever seen.
Adam B. Green, InfoWorld

Reflex excels as an analytical tool . . . this program can become everyman's database manager.
Frank J. Derfler, PC Magazine

Borland has done it again.
Sheldon L. Richman, Washington Post"

YES! I want the best

Send me Reflex and Reflex Workshop at:

$199.95*

To order by phone, or for a dealer nearest you,

call (800) 255-8008

in CA call (800) 742-1133

____	Reflex/Workshop(s) at $199.95*	$ ____
____	Reflex at $149.95*	$ ____
____	Workshop at $69.95*	$ ____
____	Upgrade to 1.1 and Workshop at $59.95	$ ____
____	Upgrade to 1.1 at $10.00 (you must return your disks)	$ ____
	Outside USA add $10 per copy	
	CA and MA res. add sales tax	$ ____
	Amount enclosed	$ ____

Prices include shipping to all U.S. cities
†You must have an IBM or true compatible running DOS 2.0 or later.
My computer's name and model is: ____________

The disk size I use is: ☐ 3½" ☐ 5¼"
Payment: VISA MC Bank Draft Check
Credit card expiration date ___/___
Card # ____________

NOT COPY PROTECTED
****60-DAY MONEY-BACK GUARANTEE**

Name: ____________
Shipping Address: ____________
City: ____________
State: ____________ *Zip:* ____________
Telephone: ____________

CODs and purchase orders WILL NOT be accepted by Borland. Outside USA make payment by credit card or International Postal Money Order.

**Limited time offer until September 1, 1986*

***YES, if within 60 days of purchase this product does not perform in accordance with our claims, call our customer service department and we will gladly arrange a refund.*

†Minimum System Requirements:
384K—Runs on IBM PC, AT, XT, and true compatibles. IBM Color Graphics Adapter, Hercules Monochrome Graphics Card or equivalent.

Reflex works with Intel's Above Board AT and Above Board/PC, AST's RAMpage! and RAMpage! AT, Quadram's Liberty-PC and Liberty-AT, Tecmar's 640 Plus, IBM's EGA and 3270/PC, AT&T's 6300, and many others.

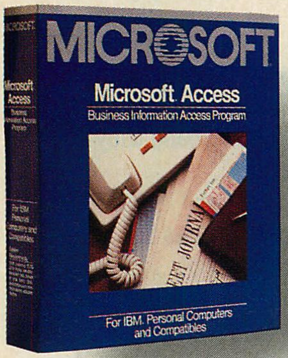
R6

BORLAND INTERNATIONAL

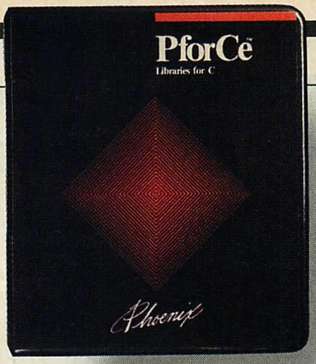
4585 SCOTTS VALLEY DRIVE
SCOTTS VALLEY, CA 95066
(408) 438-8400 TELEX: 172373

Vive la différence

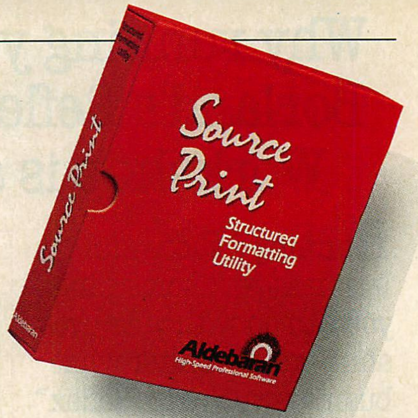
CIRCLE NO. 254 ON READER SERVICE CARD



Microsoft Access version 1.01



PforCe library from Phoenix



Aldebaran Laboratories' Source Print

Intel Corporation announced **DOS PSCOPE**, a software debugging tool that allows users to observe program execution at the source code level by using high-level procedure and variable names and labels. It provides software engineers with a symbolic test environment for program execution and includes source code display, high-level code patching, and a procedural command language. $995; in quantities of two or more, $795 each.
Intel Corporation, Literature Dept., W280, 3065 Bowers Avenue, Santa Clara, CA 95051; 800/548-4725
CIRCLE 321 ON READER SERVICE CARD

V-EMM, the Virtual Expanded Memory Manager, from **Fort's Software** is a software package that provides a true virtual memory capability to many programs that support the Lotus/Intel/Microsoft expanded memory specification. V-EMM provides the illusion that more expanded memory than the amount actually installed is available. Up to 8MB of virtual expanded memory can be provided with only a modest amount of real expanded memory. $89.95.
Fort's Software, P.O. Box 396, Manhattan, KS 66502; 913/537-2897
CIRCLE 322 ON READER SERVICE CARD

Source Print, a structured formatting utility that organizes data and clarifies print-outs, has been announced by **Aldebaran Laboratories, Inc.** This utility for database and computer programming can format print-outs of source code in C, Pascal, BASIC, and dBASE II and III. A unique structure outlining feature draws lines on a print-out to indicate to programmers at a glance the overall structure of a program. A print feature provides automatic indentation based on block nesting levels. $97.
Aldebaran Laboratories, Inc., 3738 Mt. Diablo Blvd., Building 312, Lafayette, CA 94549; 415/283-7084
CIRCLE 326 ON READER SERVICE CARD

A comprehensive library of object-oriented functions and subsystems for the C programming language has been introduced by **Phoenix Computer Products Corporation**. **PforCe** is written in C and assembly language and offers programmers both high- and low-level functions that are fully integrated, optimized, debugged, and ready to use. High-level functions allow programmers to manipulate windows, screens of fields, and Lotus-like menus, and handles databases as objects. Low-level functions give programmers complete hardware control and defaults that can be changed. PforCe is available for the Microsoft, Lattice, Computer Innovations, and Wizard C compilers. $395.
Phoenix Computer Products Corporation, 320 Norwood Park S, Norwood, MA 02062; 617/762-5030
CIRCLE 323 ON READER SERVICE CARD

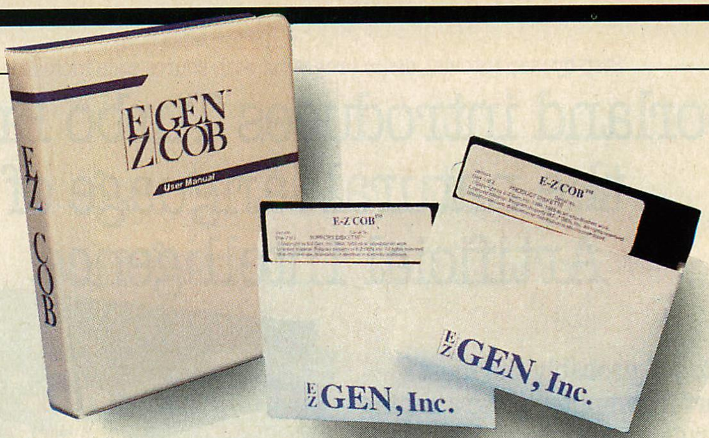
A software product for computer-aided mathematics has been introduced by **MathSoft, Inc.** With **MathCAD**, users can enter and calculate equations, create plots, and enter and edit text on the same screen. MathCAD's simple word processor style permits the user free-form entry, text-like editing of equations, familiar keystrokes, and equations that appear on the screen just as they would appear in textbooks or on a blackboard. Single keystroke computations result as a single number or plot, and users can print as a document all information as it appears on the screen. MathCAD understands the equations and automatically presents them in correct form on screen; it sizes brackets and fraction bars interactively as the user enters an equation and places exponents, subscripts, square roots, and summation signs in arbitrarily complex combinations. $189.
MathSoft, Inc., One Kendall Square, Cambridge, MA 02139; 800/MathCAD; in Massachusetts, 617/577-1017
CIRCLE 324 ON READER SERVICE CARD

Microsoft Access version 1.01 is available from **Microsoft Corporation**. New features of this business information access program for electronic communications include updated custom menus, a new installation feature for hard disks, speed enhancements, and non-copy-protected disks. $250.

Also from Microsoft comes a networking product for its XENIX System V/286 operating system. **Microsoft Networks** for XENIX provides networking and distributed file system capabilities on computers running XENIX System V and transparent file sharing with PCs running DOS in a LAN. Microsoft Networks is being offered to OEMs licensed for XENIX System V. Pricing is variable; OEMs should contact the company.
Microsoft Corporation, 16011 N.E. 36th Way, Redmond, WA 98052-6399; 800/426-9400
CIRCLE 325 ON READER SERVICE CARD

Ungermann-Bass, Inc. introduced a software package that enables PCs to communicate with IBM and Digital Equipment Corporation (DEC) host applications programs, Microsoft Network-based file and print servers, and PC applications via the Ungermann-Bass Net/One general purpose LAN. The **3270 Personal Connection** operates with the Net/One Personal Connection Network Interface Unit and the Net/One NIU-74 to connect PCs to any IBM 3274 controller. When running **Model I** of the program, the PC emulates an IBM 3278/79 terminal; a hot key permits toggling between a host session and a PC session. **Model II** enables a user to connect to multiple IBM host systems, viewing up to four applications simultaneously, or to connect to a DEC host to run two VT100 terminal sessions at the same time. Model I, $95; Model II, $595.
Ungermann-Bass, Inc., 2560 Mission College Blvd., Santa Clara, CA 95052; 408/496-0111
CIRCLE 329 ON READER SERVICE CARD

C compiler by Tartan Laboratories



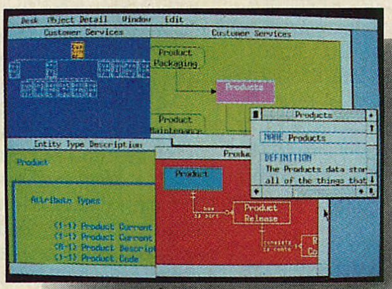
E-Z COB by E-Z GEN

Dynamical Systems, Inc. has released a software utility intended to speed up the PC/AT. The **AT SpeedFixer Plus** eliminates the disk drive errors that occur on ATs equipped with frequencies above 6 MHz. A keyboard speed-up utility reprograms the chip inside the AT keyboard to make it faster and more responsive. The AT SpeedFixer also provides software for speeding up the AT hard disk by 50 percent (by changing the interleave). $39.95.
Dynamical Systems, Inc., 2511 Fulton Street, Berkeley, CA 94704; 800/227-2400, ext. 929
CIRCLE 328 ON READER SERVICE CARD

E-Z GEN has announced **E-Z COB**, a programmer productivity tool capable of generating COBOL programs for use on the PC as well as for code generation and prototyping of host on-line systems, such as IMS, CICS, and 8180/DPPX. Features of E-Z COB include a powerful full-screen painting facility that supports both monochrome and color CRTs, a reusable code library that permits the storage of COBOL macros to be used by E-Z COB, user exits that enable a developer to include his own code in the programs, and context-sensitive help and tutorials during the development process. E-Z COB is designed for use with REALIA COBOL. $495.
E-Z GEN, Inc., 1019 Mt. Pleasant Way, Cherry Hill, NJ 08034; 609/428-0211
CIRCLE 335 ON READER SERVICE CARD

Concept Development Systems, Inc. has announced **Line Plus**, an asynchronous telecommunications software system for computer-to-computer communications and remote access. File transfer protocols include Reliable, XMODEM, SmartCom, and standard text protocols. Emulation includes ANSI standard with full color, IBM 3101, DEC VT100/102, DEC VT52, DG Dasher D220 with full color, Televideo 912, LSI ADM3A, TTY, and a special 25-line color terminal. Line Plus also enables a user to connect an office PC to a PC at home. Single user version, $199. Site and corporate licenses available.
Concept Development Systems, Inc., 2778 Hargrove Road, Suite 349, Smyrna, GA 30080; 404/434-4813
CIRCLE 334 ON READER SERVICE CARD

A new integrated systems development environment for planning, analyzing, designing, and constructing computer-based information systems has been announced by **KnowledgeWare, Inc.** The **Information Engineering Workbench** software family uses expert-system and CAD/CAP techniques to automate information engineering. A mouse



Information Engineering Workbench screen

creates, verifies, and revises full-color diagrams on the computer screen. Not only the picture, but also the meaning of the diagram is stored in a knowledge base called the Encyclopedia. The user can view different diagrams or portions of the diagrams simultaneously through the use of windows. $7,500.
KnowledgeWare, Inc., 16250 Northland Drive, Suite 318, Southfield, MI 48075; 313/443-0410
CIRCLE 331 ON READER SERVICE CARD

Prelude is a distributed information system that provides realtime data management in a mixed network of minicomputers and desktop computers or remote processors. Created by **VenturCom, Inc.**, this system is built around a powerful relational database system. Prelude's architecture permits user access to shared files anywhere on the system, whether the files are physically maintained on a PC disk drive, a file server, or a minicomputer tape drive. Two-user configuration license for PC/XTs, $1,800; configuration of 16 PC/XTs linked with a VAX supporting another 16 terminals, $28,000.
VenturCom, Inc., 215 First Street, Cambridge, MA 02142; 617/661-1230
CIRCLE 330 ON READER SERVICE CARD

From **Tartan Laboratories, Inc.** comes a highly optimizing **C Compiler** for the RT/PC. It runs under IBM Academic Information System's 4.2 operating system and will be available for the IBM Advanced Interactive Executive (AIX) operating system. $1,000.
Tartan Laboratories, Inc., 477 Melwood Avenue, Pittsburgh, PA 15213; 412/621-2210
CIRCLE 327 ON READER SERVICE CARD

A microcomputer diagnostic product has been announced by **Windsor Technologies, Inc.** Version 1.06 of **PC-Technician** performs extensive diagnostic testing on a system's overall hardware as well as on the its individual components. The package includes test materials and a carrying case. $200.
Windsor Technologies, Inc. 66 Bovet Road, Suite 380, San Mateo, CA 94402; 415/345-5700
CIRCLE 333 ON READER SERVICE CARD

Erratum: The photo spread at the top of page 30 in the May 1986 Tech Releases was captioned incorrectly. The photograph shows a new line of internal tape and disk/tape subsystems by **Tallgrass Technologies Corp.**

The material that appears in Tech Releases is based on vendor-supplied information. These products have not been reviewed by the PC Tech Journal *editorial staff.*

Turbo Pascal® Source Code Included

Borland's new Turbo Editor Toolbox, "Best of the Year" award winner, lets you build your own word processor for only $69.95!

Turbo Editor Toolbox™ lets you build the best of all word processors into your own word processor. All the modules, techniques, instructions, and Turbo Pascal source code are at your fingertips. You'll quickly learn how to integrate editor procedures and functions into your programs, or you can use Turbo Editor Toolbox "as is," because it has everything.

You get Turbo Pascal source code and everything you need to build your own word processor

The modules, the manual, ready-to-compile source code, and a full-featured word processor called MicroStar,™ which we probably ought to sell separately because it's an excellent text editor. But anyway, you get it *free* as part of our new Turbo Editor Toolbox. (Maybe this is why Jerry Pournelle of BYTE magazine recently wrote that *"Borland International is a public benefactor. The company continues to pour out good, well-documented products at reasonable prices."*) Your free MicroStar includes a complete pull-down menu user interface which you can use "as is," or you can modify it for inclusion in your Turbo Pascal programs.

As well as MicroStar, you also get a complete editor ready to include in your programs. Windows, block commands, and memory-mapped screen routines come with it.

How to turn good stuff into great stuff—maybe even green stuff!

With your new Turbo Editor Toolbox, you can make WordStar® behave like MultiMate.™ You can support windows just like Microsoft's® Word. And do it as fast as WordPerfect® does it. In other words, you can do what they should have done. You just go in there, tinker, fiddle, fool around, and come up with your own version—which will be the best word processor you've never seen before. (And if you want to sell it, go for it; we're not the kind of company that'll send bean-counters and ambulance-chasers after you for royalties.)

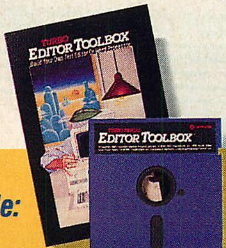
How to do windows without jamming your fingers back in your wallet

State-of-the-art "windowing" techniques are part of our new Turbo Editor's repertoire. Sophisticated but easy-to-learn techniques let you design your word processor to show several documents—or several parts of the same document—all at once.

Turbo Editor Toolbox lets you open the windows you want—wherever you want them—at a price that won't make you want to jump out of them.

You get a lightning-fast editor, innumerable features, and a 60-day money-back guarantee** for only $69.95

For only $69.95, you can build your own word processor and make it do whatever you want it to do. This already popular new program is just one more way that Borland helps you help yourself. So call us or the dealer nearest you. All the telephone numbers and ordering information are in the adjacent coupon.



Standard Turbo Editor Toolbox features include:

- Wordwrap
- UNDO last change
- Auto-indent
- Find & Find/Replace with options
- Set left and right margins
- Block mark, move and copy
- Tab, insert and overstrike modes, centering, etc.
- Multiple windows
- Multitasking
- RAM-based editor
- Paging, scrolling and text display

"The new Turbo Editor Toolbox is the Turbo Pascal source code to just about anything you ever wanted a PC-compatible text editor to do, along with a really excellent book of instructions on what text editors are and how to use the Toolbox to build a custom text editor . . . you can't afford to be without this."

Jerry Pournelle, BYTE Magazine, discussing Turbo Editor Toolbox, to which he gave his "Best Of The Year" Award

YES! I want the best

Rush me Turbo Editor Toolbox at:

$69.95

To order by phone, or for a dealer near you,
call (800) 255-8008
in CA call (800) 742-1133

Send me ____ Turbo Editor Toolboxes at $ ______
Outside USA add $10 per copy
CA and MA res. add sales tax $ ______
Amount enclosed $ ______
Prices include shipping to all U.S. cities
Payment: VISA MC Bank Draft Check
Credit card expiration date ___/___
Card # |_|_|_|_|_|_|_|_|_|
|_|_|_|_|_|_|_|_|_|

†You must have an IBM or true compatible running DOS 2.0 or later.
My computer's name and model is: ____________

The disk size I use is: ☐ 3½" ☐ 5¼"

NOT COPY PROTECTED
****60-DAY MONEY-BACK GUARANTEE**

Name: ____________
Shipping Address: ____________
____
City: ____________
State: ____________ Zip: ____________
Telephone: ____________

CODs and purchase orders WILL NOT be accepted by Borland. Outside USA make payment by credit card or International Postal Money Order.

**YES, if within 60 days of purchase this product does not perform in accordance with our claims, please call our customer service department and we will gladly arrange a refund.

†Minimum System Requirements: 192K
Runs on IBM PC, XT, AT, PCjr., and true compatibles

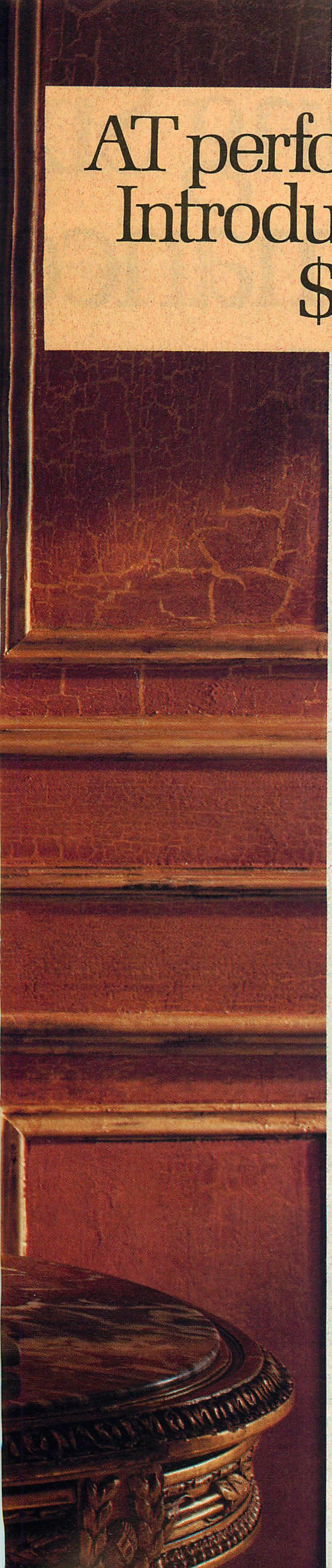
TE6

BORLAND
INTERNATIONAL
Vive la différence

4585 SCOTTS VALLEY DRIVE
SCOTTS VALLEY, CA 95066
(408) 438-8400 TELEX: 172373

CIRCLE NO. 255 ON READER SERVICE CARD


TELECAT 286
110
100
90
80
70
60
50
40
30
20
10
0
PC/AT
TELECAT 286
PC/XT
Size Performance Price
TeleVideo
TeleVideo



AT performance at an XT price. Introducing the TeleCAT-286. $2995. Complete.

With TeleVideo, you always settle for more.

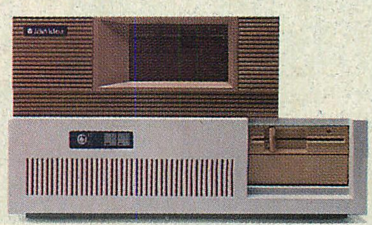
Up till now, with a mid-range budget, you had to settle for mid-range performance. And a mid-range set of features.

But not anymore. Because now, there's the new TeleCAT-286,™ from TeleVideo. An IBM AT-compatible machine that lets you settle for an entirely new concept in medium-priced PCs: more.

More Performance.

The TeleCAT-286 retails for $2995, roughly the same as a comparably-equipped IBM XT. But the similarity ends there. Instead of starting you off with a stripped-down box, we've loaded up the TeleCAT-286 with everything you need. Like a 20MB hard disk. A 1.2MB floppy. An Intel 80286 CPU that runs at either 6 or 8 MHz clock speed. There's even a high-resolution monitor for text and graphics.

28% Smaller Footprint:
What you do with the extra desk space is up to you, but as you can see here, the TeleCAT-286 gives you a lot more of it than the IBM AT.



To make even better use of internal space, we socketed the TeleCAT-286 for one MB of RAM, and also included serial and parallel ports on the motherboard. As a result, we can still give you three extra expansion slots.

More Productivity.

Using our experience in building terminals and systems for 750,000 users worldwide, we've designed a machine that's the last word in ergonomics. With sculptured keycaps on a high-quality keyboard. LEDs on the three critical locking keys. And a footprint that's 28% smaller than the IBM AT's. So you get more of your desk back, too.

LEDs On Locking Keys:
For maximum visibility, we put our LEDs right on top of the three critical locking keys, so they won't get covered up by overlays.

Find Out Even More: 1 (800) TELECAT.

There's a whole lot more we could tell you about the TeleCAT-286. But it's an even better idea to get your hands on it. So call us at 1 (800) TELECAT, Dept. 202, and we'll tell you the nearest place you can try one.

The TeleCAT-286. Our 20MB version is $2995; 30MB, $3495. For high performance at a low price, don't settle for less.

TeleVideo®
Settle for more.

TeleVideo Systems, Inc. 1170 Morse Avenue
Sunnyvale, California 94088-3568 • (408) 745-7760

CIRCLE NO. 182 ON READER SERVICE CARD

Introducing Six
Finally, Performance

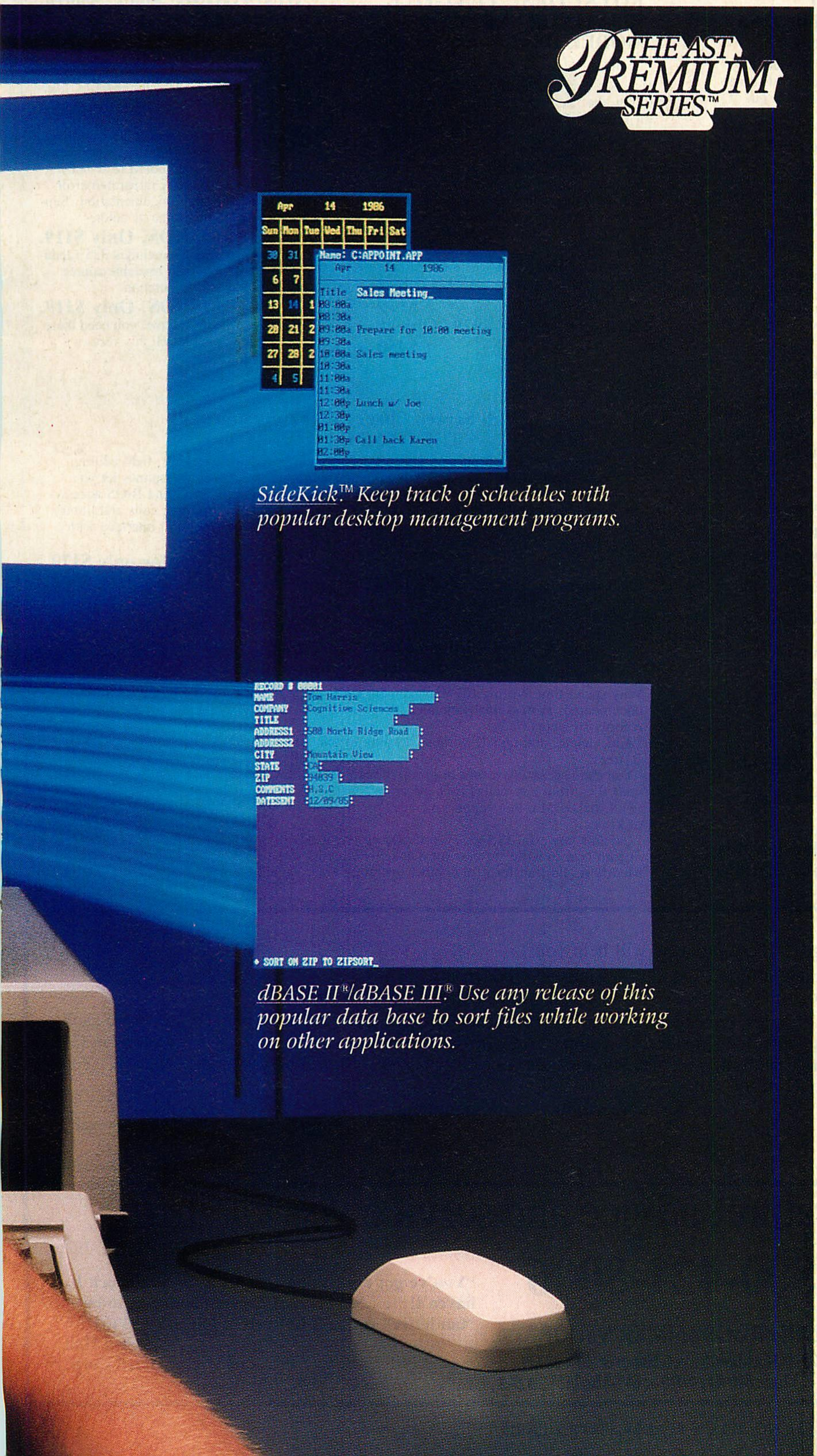
WordStar.® Write a letter while running up to eight other applications.

Lotus 1-2-3.® All releases of 1-2-3 run in SixPakPremium's DESQview™ windows.

Dow Jones.® Monitor on-line stock quotations while working on other projects.

SixPakPremium.™ Exceeds Demand.

Also Compatible With Original IBM® PCs & XTs

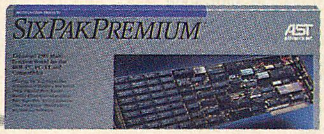


SideKick.™ *Keep track of schedules with popular desktop management programs.*

dBASE II®/dBASE III.® *Use any release of this popular data base to sort files while working on other applications.*

SixPakPremium, the new generation multifunction board designed for new generation IBM XTs, is here! Now one person and one PC have the power to meet the demands of business. AST's newest solution for new XTs, existing PCs, XTs and compatibles, SixPakPremium offers all the popular multifunction features plus two megabytes of expanded memory and the powerful software to use it.

Premium Performance. Now, run up to nine applications simultaneously. Sort a data base, write a letter, calculate a spreadsheet, monitor stock quotations and print labels...all at the same time...on the same PC! And, our windows let you view applications instantly.



SixPakPremium is fully compatible with all current PC applications, including expanded memory applications such as Lotus 1-2-3 Release 2.0. And DESQview, our multitasking/windowing environment is compatible with more than 200 of the most popular packages.

Premium Features. SixPakPremium offers up to two megabytes of expanded memory, two serial ports, parallel port, game port, battery-backed clock/calendar, DESQview and SuperPak™ utilities.

Premium Quality. As the leader in PC enhancement, with over a million products shipped, AST is known worldwide for its highly reliable products.

Make The Premium Choice. For performance that exceeds demand, call our Customer Information Center **(714) 863-1333.** Or send the coupon to AST Research, Inc., 2121 Alton Avenue, Irvine, CA 92714.

AST RESEARCH INC.

CIRCLE NO. 110 ON READER SERVICE CARD

Yes, I want to learn more about Premium Performance. Please send me your special SixPakPremium information package today!

Name__________

Title__________

Company__________

Address__________

City________ State______ Zip______

Phone (   )__________

CIRCLE NO. 220 ON READER SERVICE CARD

ERIC M. MILLER

RTDs and Thermocouples

Two approaches to measuring temperature with a data acquisition board work for different range and accuracy requirements.

Temperature is the foremost environmental factor affecting most processes, and it is the most measured. Two different approaches to temperature measurement with a data acquisiton board make use of platinum resistance temperature detectors (RTDs) and thermocouples.

RTDs are highly accurate in the range –200 to +600 degrees Celsius. Accuracies to 0.1 degree Celsius are obtainable, and repeatability can be within .01 degree. Most RTDs have a nominal 100-ohm resistance at 0 degrees Celsius and a temperature drift of .385 ohms per degree. In figure 1, an RTD is stimulated by a 2-milliampere current source. The voltage produced is amplified by a factor of 10 to put it into a convenient range for most data acquisition cards. Users are usually inclined to increase the current source to provide higher voltage output with temperature, thus obviating the need for the instrumentation amplifier. Unfortunately, this will cause self heating in the platinum RTD and result in possible inaccuracies in any measurements that are taken.

The RTD's resistance is a mildly nonlinear function of temperature and adheres to a third-order polynomial. Final temperature determination is handled most easily by piecewise linear approximation in software, based upon the values given in the device's table by the RTD manufacturer.

Thermocouples are used primarily for instrumenting wide ranges or the high ends in temperature measurement. Tungsten thermocouples (types C and G) can sense temperatures to 2,300 degrees Celsius. More common types (such as type K) adequately cover the span of 0 to 1,250 degrees Celsius, with accuracies to 1 degree Celsius.

The thermocouple responds to a temperature differential between the sensing end and the reference end. The voltage produced (called the Seebeck voltage) assumes that the reference junction is at 0 degrees Celsius. An electronic circuit provides an appropriate correction voltage based on the temperature of the reference end of the thermocouple to give cold-junction compensation.

A drawback to thermocouples is that the Seebeck voltage is low and ranges from 6 to 60 microvolts per degree Celsius, thus the total output of the thermocouple is rarely above 70 to 80 millivolts (mV). To interface a thermocouple to a data acquisition card, the user provides the cold-junction compensation and sufficient gain. Figure 2 shows one approach using an Analog Devices AD595. The AD595 incorporates cold-junction compensation and gain and fault indication in a single IC. Cold-junction compensation is accomplished by sensing the temperature at the IC and applying a correction factor to its output voltage. The nominal output voltage of the AD595 is 10 mV/degree Celsius. The Seebeck voltage is not constant over a thermocouple's useful range. Final linearization of the output can take place in the user's program in two ways: the software may take in the circuit's voltage and convert it to temperature via a polynomial (typically eighth order) that approximates the thermocouple's output, or a piecewise linear approach can be used. The appropriate tables are available in reference books for thermocouples.

Eric M. Miller is president of Miller Technology, Inc., a firm that specializes in analog-to-digital hardware/software systems.

FIGURE 1: *The RTD Method*

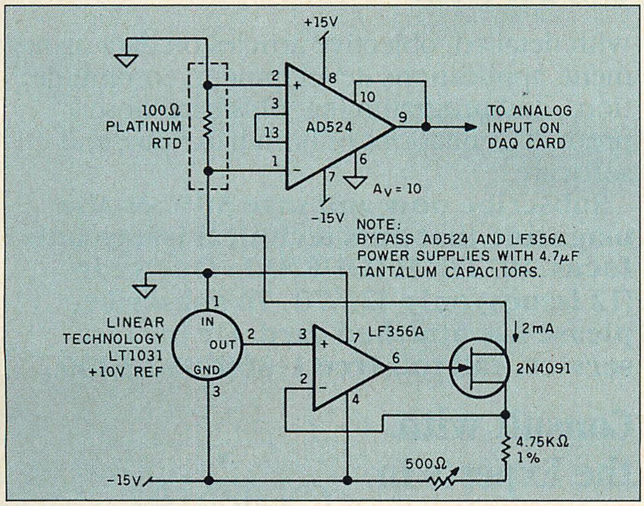
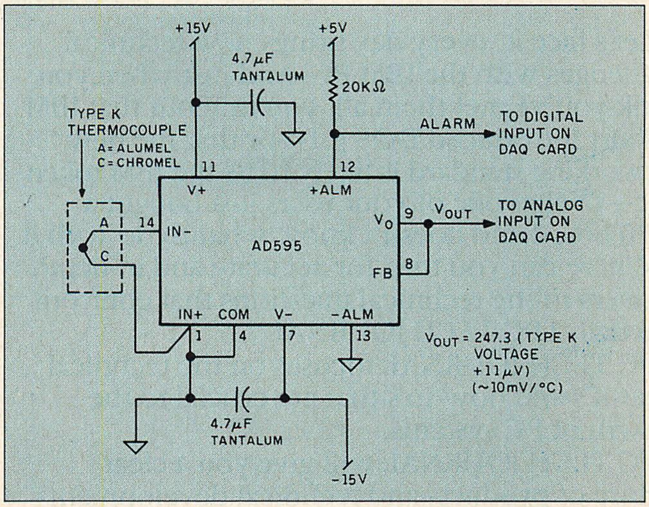


FIGURE 2: *A Thermocouple Approach*



productivity TOOLS

It's easier to do a job right when you have the right tools. That's old wisdom, but what was true yesterday is still true today. Your time is valuable and we at Programmer's Connection are ready to help by offering you some of the latest and most advanced productivity tools available. Here's a closer look at some of them.

Brief by Solution Systems — 195 CALL
Powerful, customizable, programmer-oriented text editor that features a full undo, automatic indentation for the C language and the ability to run programs in DOS from within the editor. It uses all the memory available to it and can edit files with unlimited size in its multiple windows. The compiled macro language has a rich set of commands. It supports regular expression searches, horizontal scrolling, mnemonic key assignments, keystroke macros and keyboard reconfiguration. It comes with extensive documentation, a tutorial and an on-line help system.

CodeSmith-86 by Visual Age — 145 109
Powerful multi-window program debugger with source code display windows that allow you to step and scroll your C, Fortran and Pascal source code. You can use its dual-mode patching assembler to instantly change instructions right on the display or its branch-to-patch mode to insert code between existing instructions. It supports stop on data compare/miscompare, breakpoints and passpoints, dual monitors and much more. It comes with a self-contained operating system that allows you to debug any code — even DOS itself.

Epsilon by Lugaru Software — 195 165
Extremely fast and powerful Emacs-like macro text editor that provides a complete programming environment. It features the ability to compile programs in one window while concurrently editing others in other windows. Now you don't have to wait for your compiler to finish before you can begin making your corrections! Powerful macro commands give you virtually unlimited editing capabilities. Other features include customizable keyboard, keyboard macros, automatic swap file, regular expression search, unlimited number and sizes of files, language support for C and others, recovery of deleted text (undo n times) and more. A very fast, context sensitive, on-line help facility is also included. Requires 256K memory.

Microsoft Macro Assembler (MASM) — 150 99
Very fast, standard PC macro assembler that fully supports the 8088, 80186 and 80286 instruction sets and their corresponding math coprocessors. It's completely upward compatible with the IBM Macro Assembler and comes complete with a fast MS-DOS overlay linker, a library manager, a MAKE utility, a cross reference utility and excellent documentation. Additional utilities include EXEPACK, an EXE file compression utility and EXEMOD, an EXE file header utility. Also featured is SYMDEB, an interactive symbolic program debugger that supports source-level debugging of Microsoft C, Fortran and Pascal programs. It supports source line single-stepping and breakpointing, full I/O redirection, stack tracing and shell escapes for executing DOS commands.

PERISCOPE by Data Base Decisions

Periscope I	295	249
Periscope II	145	115
Periscope II-X	99	85

Full featured resident symbolic debugger that features in-line symbolic assembly, debugging with high-level source code and the ability to view text files while debugging. It supports most languages, dual screens, graphics, comprehensive breakpoints, crash recovery and much more. There are three configurations of Pericope available. Periscope I comes with an NMI breakout switch and a crash-proof write-protected memory board (for IBM PC's or close-compatibles only). Periscope II comes with an NMI breakout switch but does not include the write-protected memory board. Periscope II-X is a software only version that comes without either the NMI breakout switch or memory board. The software for each configuration is the same. Requires 128K memory.

Pfantasy Pac by Phoenix Software — 1295 895
Combination of programmer's productivity tools that includes six valuable Phoenix products: Plink86 Plus, Pfix86 Plus, Pfinish, Pmaker, Pmate and Ptel. Plink86 Plus is an extensive overlay linker. Pfix86 Plus is a symbolic debugger that supports Plink86 overlays. Pfinish is an execution profiler that finds the inefficient parts to your programs. Pmaker is a make facility that automates the process of creating new programs after changes have been made. Pmate is a powerful macro text editor. Ptel is a telecommunications program for binary file transfers. Please turn the page and refer to the Phoenix Products section for more information.

PolyMake by Polytron Software — 99 79
Intelligent program builder and maintenance tool for MS-DOS that automates the process necessary to create new versions of a program. You can use its built-in rules to rebuild your programs or specify your own. Then, whenever you make changes to your source code and are ready to recompile, you just invoke PolyMake. It determines which source files need compilation and automatically creates the new program using the minimum number of operations necessary to bring the program up to date. It works with source files in any language and can be used to maintain non-program files as well. It uses the UNIX make syntax and can run UNIX makefiles. 256K memory recommended.

Source Print
by Aldebaran Laboratories — New 139 119

Source formatting and printing utility for C, Pascal, BASIC, dBase and Modula-2 that indexes and cross references your progams for you. It provides page headings, line numbers, a table of contents, automatic indentation and extensive printer control. You can have it draw lines connecting the beginning and end of nested control structures to make even the most complex program structures stand out clearly. Key words may be emphasized with printer styles like boldface and underline. You can extract and print by line number range or by selected subtitled procedures and functions from one or more source files. Requires 192K memory.

SRMS Software Revision Mgmt System
by Quilt Computing — New version 125 109

Set of integrated utilities that maintains a history of the revisions made to a program. You have complete access to all versions of the program, although any source code that is common to different versions are not duplicated. It supports any type of source files including C, Fortran, BASIC, Pascal, COBOL and assembly code. There are extensive provisions for including comments and notes describing each version. Any version of the program may be quickly and easily restored with a single command. It can be used interactively or in batch environments. Requires 128K memory.

XTC by Wendin — 99 84
Programmer's editor that features a complete, structured, multitasking macro language that may be interpreted or compiled. It supports up to eight windows and twenty text buffers for holding files and blocks of text. It has the ability to edit unlimited-sized files and allows you to compile from within one window and have output redirected into another. Additional features include undo N times, removal of Wordstar high bits, tab expansion and compression, auto-indentation and the ability to edit extra long lines. The command structure and cursor movement controls are customizable. Includes complete editor source code in Microsoft Pascal. Requires 256K memory.

CALL TOLL FREE

800·336·1166
U.S. OHIO 216-877-3781

800·225·1166
CANADA

programmer's connection

Be sure to turn the page for the largest advertised selection of programmer's development tools for IBM Personal Computers and compatibles.

PROGRAMMER DEVELOPMENT TOOLS FOR THE IBM-PC/XT/AT
and compatibles.

NO
*Shipping Charge**
Handling Charge
Insurance Charge
Credit Card Charge
C.O.D. Charge
Purchase Order Charge
Hidden Charges
PLUS OUR NO RISK GUARANTEE

*When shipping via standard United Parcel Service.

apl language

	LIST	OURS
APL*PLUS/PC System by STSC ... *New version*	595	**449**
APL*PLUS/PC Tools Vol 1 by STSC	295	**239**
APL*PLUS/PC Tools Vol 2 by STSC	85	**69**
APL*PLUS/UNX System *For AT Xenix* by STSC	995	**795**
Btrieve *ISAM File Manager* by SoftCraft ... *New version*	250	**195**
Financial/Statistical Library by STSC	275	**219**
FRESCO *Business Graphics Library* by Mr. APL	300	**269**
Pocket APL by STSC	95	**79**
STATGRAPHICS by STSC	695	**539**

artificial intelligence

	LIST	OURS
ExpertEASE by Human Edge ... *New*	695	**589**
ExpertEDGE by Human Edge ... *New*	795	**659**
Experteach *Complete System* by Intelliware	475	**389**
EXSYS *Expert System Development Software* by EXSYS	395	**339**
GCLISP *Golden Common LISP* by Gold Hill ... *All models*	CALL	**CALL**
Insight I *AI Primer* by Level Five Research	95	**75**
Insight II by Level Five Research	485	**389**
Microsoft LISP *Common Lisp*	250	**189**
Methods *Smalltalk-based Prototyping* by Digitalk	250	**209**
MicroProlog by Programming Logic Associates ... *New*	250	**219**
with APES	425	**369**
Professional MicroProlog by Programming Logic ... *New*	395	**349**
with APES	650	**569**
Prolog-86 from Solution Systems	125	**CALL**
Prolog-86 Plus from Solution Systems	250	**CALL**
QNIAL *Combines APL with LISP* by NIAL Systems ... *New*	375	**359**
Small-X by Kaplan	125	**99**
TransLISP from Solution Systems ... *New*	75	**CALL**
Turbo Prolog Compiler by Borland International ... *New*	100	**CALL**

assemblers and debuggers

	LIST	OURS
8088 Assembler *w/Z-80 Translator* by 2500 AD	100	**89**
Advanced Trace-86 *with ASM Interpreter* by Morgan	175	**139**
Codesmith-86 Debugger by Visual Age	145	**109**
Cross Assemblers from 2500AD ... *Over 25 varieties*	CALL	**CALL**
Microsoft Macro Assembler *with utilities*	150	**99**
Periscope I by Data Base Decisions	295	**249**
Periscope II by Data Base Decisions	145	**115**
Periscope II-X by Data Base Decisions ... *New*	99	**85**
The PROFILER *with Source Code* by DWB Associates	125	**95**
Turbo EDITASM *Fast Assembler* by Speedware	99	**84**
Visible Computer: 8088 by Software Masters	80	**65**

basic language

	LIST	OURS
BetterBASIC by Summit Software ... *Now BASICA Compatible*	200	**165**
8087 Math Support	99	**85**
Btrieve Interface	99	**85**
C Interface ... *New*	CALL	**CALL**
Run-time Module	250	**225**
Microsoft QuickBASIC Compiler ... *BASICA Compatible*	99	**79**
Professional BASIC by Morgan Computing	99	**79**
8087 Math Support	50	**47**
True Basic from Addison-Wesley	150	**105**
Run-time Module	500	**435**

blaise products

	LIST	OURS
Asynch Manager *Specify for C or Pascal*	175	**139**
C Tools Combination Package ... *Both Items Below*	175	**149**
C Tools	125	**105**
C Tools 2	100	**84**
Exec *Program Chainer*	95	**79**
Pascal Tools Combination Package ... *Both Items Below*	175	**149**
Pascal Tools	125	**105**
Pascal Tools 2	100	**84**
Turbo ASYNCH *for Turbo Pascal*	100	**84**
Turbo POWER TOOLS *for Turbo Pascal*	100	**84**
View Manager *Specify for C or Pascal*	275	**209**
with Source Code ... *Special Price*	295	**239**

borland products

	LIST	OURS
REFLEX *Data Base System* ... *New*	99	**75**
Turbo DATABASE TOOLBOX	55	**38**
Turbo EDITOR TOOLBOX	70	**54**
Turbo GAMEWORKS TOOLBOX	70	**54**
Turbo GRAPHIX TOOLBOX	55	**38**
Turbo LIGHTNING	99	**75**
Turbo PASCAL	70	**49**
with 8087 or BCD	110	**77**
with 8087 and BCD	125	**84**
Turbo Prolog Compiler ... *New*	100	**CALL**
Turbo TUTOR *for Turbo PASCAL*	35	**28**

c compilers

	LIST	OURS
C-86 Compiler ... *See Computer Innovations Section*	395	**289**
Datalight C Compiler *Small Memory Model*	60	**49**
Datalight Developer's Kit *with Large Memory Model* ... *New*	99	**79**
DeSmet C Compiler *with Source Debugger*	159	**145**
Eco-C *Complete Development System* by Ecosoft	125	**89**
Lattice C Compiler ... *See Lattice Section*	500	**299**
Let's C Compiler by Mark Williams	75	**69**
with csd Source Level Debugger	150	**129**
Microsoft C Compiler *with Source Debugger* ... *New version*	495	**CALL**
MWC-86 by Mark Williams ... *Special Price*	495	**289**
Wizard C Compiler *Includes Lint* by Wizard Systems	450	**369**

c interpreters

	LIST	OURS
C-terp by Gimpel Software ... *Specify compiler interface*	300	**239**
Instant C by Rational Systems	500	**379**
Interactive C by IMPACC Associates	249	**219**
Run/C by Age of Reason	150	**99**
Run/C Professional by Age of Reason	250	**189**

c utilities

Also refer to Blaise, Computer Innovations, Lattice, Microsoft, Phoenix, Polytron, SoftCraft and Xenix System V sections.

	LIST	OURS
APT Application Programmer's Toolkit by Shaw American	395	**339**
Basic C Library by C Source	175	**135**
C Essentials by Essential Software	100	**85**
C-lib by vance info systems	195	**125**
C Power Packs by Software Horizons	CALL	**CALL**
c-tree *ISAM File Manager with source* by FairCom	395	**329**
C Utility Library by Essential Software	185	**139**
C Windows by Syscom	100	**89**
C Wings by Syscom	50	**45**
dbVISTA *Single-User DBMS* by Raima	195	**159**
with Source Code	495	**429**
dbVISTA *Multi-User DBMS* by Raima	495	**429**
with Source Code	990	**849**
EditCheck by Everest Solutions	90	**79**
Entelekon Combo Package ... *Combination of 3 items below*	200	**175**
C Function Library	130	**115**
C Windows	130	**115**
Superfonts for C	50	**45**
Essential Graphics by Essential Software ... *No royalties*	250	**219**
Flash-up Windows by Software Bottling of NY	75	**69**
GraphiC *Mono version 2.2* by Scientific Endeavors	280	**219**
GraphiC *Color version 3.0* by Scientific Endeavors	350	**299**
The Greenleaf Functions by Greenleaf Software	185	**135**
Greenleaf Comm Library by Greenleaf Software	185	**135**
The HAMMER by OES Systems	195	**175**
MetaWINDOWS by Metagraphics	185	**139**
MetaWINDOWS/Plus by Metagraphics ... *New*	235	**199**
Multi-Halo by Media Cybernetics ... *Royalties*	300	**219**
On-line Help from Opt-Tech Data Processing	149	**119**
PANEL by Roundhill ... *Library Source Available*	295	**229**
PC Lint by Gimpel Software	139	**109**
Scientific Subroutine Library for C by Peerless	175	**139**
Vitamin C by Creative Programming	150	**139**
VC Screen *Interactive Forms Designer* ... *New*	99	**85**
Zview by Data Management Consultants	245	**199**

cobol language

	LIST	OURS
Micro Focus COBOL Workbench	4000	**3599**
Micro Focus Level II COBOL	CALL	**CALL**
COMATH	200	**169**
FORMS-2	300	**269**
Level II Animator	1200	**995**
Level II SOURCEWRITER	2000	**1689**
Micro Focus Micro/SPF	175	**159**
Micro Focus Professional COBOL	3000	**2395**
Microsoft COBOL ... *See Microsoft Section*	700	**495**
Realia COBOL ... *New*	995	**795**
RM/COBOL by Ryan-McFarland	950	**675**
RM/COBOL 8X *ANSI 85 COBOL* by Ryan-McFarland ... *New*	1250	**995**

computer innovations products

	LIST	OURS
C-86 Optimizing Compiler	395	**289**
C to dBase	150	**139**
CI Probe *Source Level Debugger*	225	**199**
CI RomPac *for C-86*	195	**149**
Introducing C *C Interpreter*	125	**105**

fortran language

	LIST	OURS
ACS Time Series by Alpha Computer Service	495	**429**
Btrieve *ISAM File Manager* ... *See SoftCraft Section*		
For-Winds by Alpha Computer Service	90	**79**
Forlib-Plus by Alpha Computer Service	70	**55**
Microsoft Fortran	350	**215**
MORE FORTRAN by Peerless Engineering	125	**99**
Multi-Halo by Media Cybernetics ... *Royalties*	300	**219**
PANEL *Screen Designer* by Roundhill	295	**229**
PC Fortran Tools by Stat Com Systems	179	**159**
PolyFortran Tools by Polytron	179	**139**
RM/Fortran by Ryan-McFarland	595	**395**
Scientific Subroutine Library by Peerless	175	**139**
The Statistician by Alpha Computer Service	295	**259**
Strings & Things by Alpha Computer Service	70	**55**

Six New Shapes of BASIC

Several new BASIC interpreters combine the traditional strengths of BASICA with the advanced language features that today's programmers expect.

TED MIRECKI

During the first two decades of its existence, BASIC has experienced many transformations. It started out as an innovative mainframe language. Developed in 1964 at Dartmouth College, it was the first interactive language meant for use on terminals that were logged on to a time-sharing system. For its day, it had some advanced features: free-field input, automatic output formatting, variable-length strings, and single-step matrix operations (the last is not available in most microcomputer versions).

BASIC later played a crucial part in the microcomputer revolution and remains the most widely used microcomputer language available today. However, entirely new concepts of programming have grown up since BASIC's inception, and the language, or at least its early versions, have fallen behind the state of the art.

All IBM PCs and most compatibles come bundled with a BASIC interpreter—typically, the Microsoft product popularly known as BASICA. Its wide distribution makes BASICA the de facto standard in the microcomputer—especially the 8086—world. This fact is appreciated by programmers who distribute commercial software.

When first introduced, BASICA was a marvel of interactive programming, with graphics, string handling, event trapping, and file I/O that were both powerful and convenient. Its ability to control hardware is second only to assembly language. The implementation of graphics and sound reaches the full potential of the hardware.

However, drawing on more modern concepts of language design, implementors of other languages have fashioned products with most of BASICA's functions and a lot more besides—namely fast-executing code, structured programming constructs, and access to the full 640KB of PC memory. In the face of powerful microcomputer implementations such as Pascal and C, the continued popularity of BASICA could be considered an anachronism.

Several software companies have examined the situation and decided that it is the BASICA implementation that has been outstripped, not the BASIC language itself. They cite resilient strengths of BASIC that refuse to fade in the race toward sophisticated languages. BASIC is still the highest level language that provides the lowest level of control over hardware. It remains the *lingua franca* of the microcomputer world; more listings are published in BASIC than in any other language.

Enter the enhanced BASIC interpreters, languages that combine the traditional strengths of BASIC and the advanced features that programmers have come to expect. As they outgrow the capabilities of BASICA, many programmers look for more capable languages. While some migrate to Pascal or C, others are unwilling to make the investment in learning a new language. Still others may have a sizable amount of BASIC source code they would like to maintain. For these users, an expanded BASIC is the ideal upgrade.

The characteristics desired in a new BASIC depend on the particular need that BASICA could not fill for them. The most common are full memory utilization, execution speed, structured programming constructs, and modular program development. The

COMPUTER GRAPHIC • DOV JACOBSON

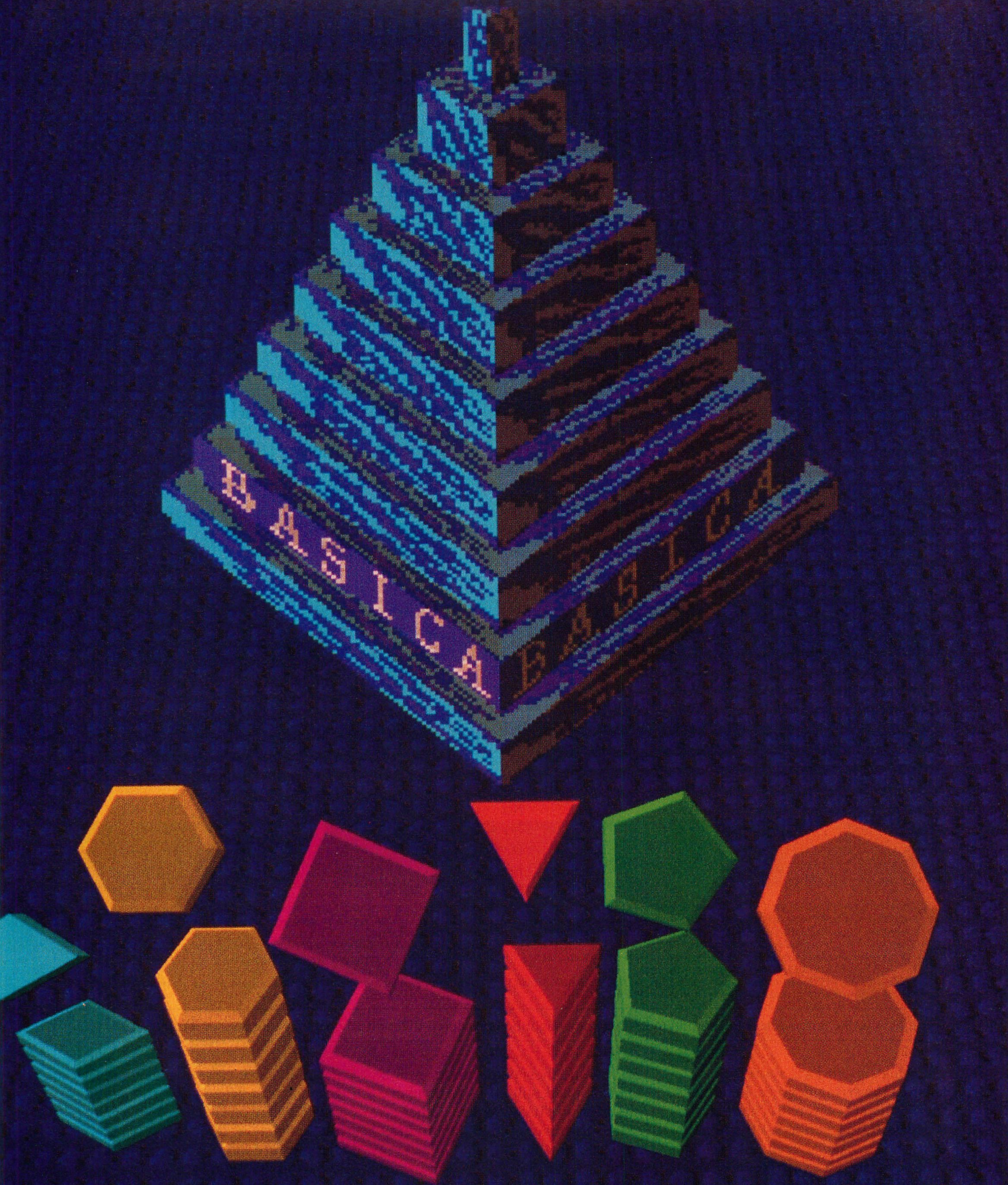

BASICA
BASICA

TABLE 1: *Features Comparison*

	BASICA	BETTER-BASIC	BUSINESS BASIC	MEGA-BASIC	PRO. BASIC	TRUE BASIC	WATCOM BASIC
Version tested	3.0	2.0	7.3	5.23	2.04	1.0	2.5
Price	Free	$199	$295	$375	$99	$149	$250
ENVIRONMENT							
Recommended memory (KB)	128	256	256	256	384	192	256
Workspace used (KB)	64	Large	64	Large	Large	Large	52
Multiple workspaces	No	Yes	Yes[a]	Yes	No	Yes	Yes[a]
Runtime package	N/A	$250	N/A	Included[b]	—[c]	$500	N/A
DOS path support	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Reads BASICA ASCII	Yes	Yes	No	No	Yes	No	No
Reads BASICA tokenized	Yes	Yes	No	No	No	No	No
Prerun syntax check	No	Yes	Yes	Partial	Yes	Yes	No
EDITOR							
Full screen	Yes	Yes	Yes[d]	No	No	Yes	Yes
Find/Replace	No	No	No	Both	Find	Both	Both
Block move	No	No	No	Yes	No	Yes	Yes
MAJOR LANGUAGE FEATURES							
Line numbers	Required	Required	Required	Required[e]	Optional	Optional	Optional
GOTO with labels	No	Yes	No	Yes	Yes	No	No
Length of variable names	40	40	32	250	308	31	31
Multiline IF	No	Yes	No	No	No	Yes	Yes
Number of loop types	2	5	1	2	2	10	5
Multiline functions	No	Yes	No	Yes	No	Yes	Yes
GOSUB with labels	No	Yes	Yes	Yes	Yes	Yes	Yes
Parameters with GOSUB	No	Yes	Yes	Yes	No	Yes	Yes
Recursive calls	No	Yes	Yes	Yes	No	Yes	Yes
Separate compilation	No	Yes	Yes	Yes	No	Yes	No
PEEK/POKE capability	Yes	Yes	No	Yes[f]	Yes	Yes	Yes
Matrix operations	No	No	No	No	No	Yes	Yes
Dynamic strings	Yes	Yes	Yes	No	Yes	Yes	Yes
Integer length (bits)	16	16	N/A	32	32	N/A	16
Real format	MS	MS	BCD	BCD	IEEE	IEEE[g]	IEEE
Max precision (digits)	16	16	14	14	16	16	16
HARDWARE SUPPORT							
8087	No	Optional	No	No	Optional[h]	Yes	Yes
Graphics	Yes	Yes	No	No[i]	Yes	Yes	Yes
Sound	Yes	Yes	No	No	Yes	Yes	Yes
Communications	Yes	Yes	No	Yes[j]	Yes[j]	Optional	Yes[j]
Light pen	Yes	No	No	No	No	No	No

[a]Multiple programs are allowed in a single workspace.
[b]A stripped-down interpreter without an editor or immediate mode is included.
[c]The manual suggests BASICA or a BASICA compiler as the runtime package.
[d]Full-screen editor provided as a utility written in BB^x.
[e]If a program is loaded without line numbers, the interpreter will add them. When saved from MEGABASIC, all programs have line numbers.
[f]The FILL and EXAMINE commands replace PEEK and POKE. Both MEGABASIC commands operate on a range of bytes, not just one byte at a time.
[g]With the 8087 only. An eight-byte format similar to IEEE is used otherwise.
[h]With the 8087 option, real numbers may be either IEEE binary or binary-coded decimal.
[i]For an extra $50, MEGABASIC will supply a graphics interface to Digital Research's GEM. The programmer must have a copy of GEM before using MEGABASIC graphics.
[j]A serial port can be read from or written to, but no equivalent of BASICA's ON COMM is provided.

The new BASIC interpreters are so dissimilar that they might be considered different languages. None is copy protected.

following six products address these features to varying degrees: BetterBASIC from Summit Software Technology; Business BASIC Extended (BB^x) from Basis Inc.; MEGABASIC from American Planning Corporation; Professional BASIC from Morgan Computing; True BASIC by True BASIC Inc.; and WATCOM BASIC from Waterloo Computing Systems.

All of these products are primarily interpreters in that they have integral editors, and their output code, even if semicompiled, still needs to be interpreted by the BASIC system. Other characteristics of each of the products are listed in table 1.

BASIC SHORTCOMINGS

BASIC is not a standardized language (see the sidebar "The ANSI Standard for BASIC"), and different implementations enhance the original BASIC in different ways—in some cases changing the language almost beyond recognition. Because the six products reviewed here are so different, directly comparing them feature-for-feature is not practical. They will first be measured against BASICA and then described separately.

Memory utilization. BASICA can use no more than 64KB of memory—a limit that applies to the *total* space for program and data, as opposed to the small

memory model of the C language, which imposes *separate* 64KB limits on program and data. Of the products reviewed here, two provide an even smaller workspace than BASICA: WATCOM BASIC, although nominally providing 64KB, takes up extra space for overhead, leaving only 52KB for the programmer's use; BB^x provides two versions of the interpreter, one with a 32KB workspace and one with 64KB. The effective capacity, however, is much less than BASICA's because BB^x's numbers are all ten bytes long, versus two, four, or eight bytes in BASICA.

The remaining four BASICs, whose workspace in shown as *large* in table 1, can use all of the PC's memory.

Structured programming. A fully structured language must provide compound (multiline) statements for decision (IF-THEN-ELSE) as well as looping, modular programming, and the localization of data within modules. BASICA provides only two multiline looping statements (FOR and WHILE). The need to string out an IF-THEN-ELSE statement on one line is, in this day and age, an unnecessary hardship. Even FORTRAN, well-known for its spaghetti code, now supports structured statements.

Professional BASIC and MEGABASIC provide no more structured constructs than does BASICA. In the case of Professional BASIC, this is understandable because it strives for total compatibility with BASICA. BB^x provides even fewer structured statements than BASICA, because it lacks the WHILE loop.

The other reviewed products add to BASICA's repertoire the multiline IF statement and several types of loops, such as DO UNTIL and DO WHILE....-UNTIL. Usually, the more types of looping constructs that a language supports, the easier it is to write readable code.

Modular programming. One of the key concepts of structured programming is the division of a program into subprogram modules. This more lucidly indicates the flow of control through a large program, and it hides messy programming details from the levels that need not be concerned with them. Subprograms are of two types: procedures (also known as subroutines), which have a large-scale effect, such as changing the value of many variables or controlling peripherals; and functions, which typically return a single value of a given numeric or string type. If a function also changes variables or performs any other actions, it is said to produce side effects.

BASICA provides a low level of subprogram capability—the same as in the original 1964 implementation of the language. Subroutines are invoked with the GOSUB statement, the target of which is a line number. The line number gives no clue to the subroutine's purpose. Furthermore, all variables are global—that is, shared between the subroutine and its caller. Writing general-purpose toolbox routines is difficult because the main program must know which variables to set in order to provide input to the subroutine, and the subroutine must know which variables to leave alone in order to avoid undesirable side effects. This can result in some long debugging sessions.

A BASICA function is defined as a single mathematical expression that must fit on one line of source code. The arguments of the function are distinct from any main program variables

> *Several software companies have examined the situation and decided that it is the BASICA implementation that has been outstripped, not the BASIC language itself.*

of the same name, and the single-line structure of the function definition protects the main program from any side effects. The high degree of isolation between the function subprogram and the calling program can, in certain situations, be as restrictive as the lack of isolation in GOSUB subroutines.

This implementation of GOSUBs and functions is the lowest common denominator provided by BASICA and adopted by the six enhanced products. The enhanced BASICs, however, provide some more advanced capabilities. Professional BASIC has made a minor improvement over BASICA by allowing named labels to be used in place of line numbers in GOSUB and GOTO statements. Procedures may thus be given descriptive names. The problem of global variables remains, however.

The other BASICs go much further in the sophistication of subprograms. Procedures are identified by name, not line number, and may have parameters the way functions do. Functions may be defined over many lines, like procedures. Parameters may be passed either by value or by reference (the former makes a temporary copy for the subprogram; the latter accesses the caller's copy, allowing a parameter value to be permanently changed). Most significantly, subprograms may be recursive—that is, they can invoke themselves.

Advanced subprograms may be internal or external. Internal subprograms are coded within the invoking program (GOSUB procedures are internal). Typically, internal subprograms and their callers share variables except for parameters and those variables declared as local. External subprograms are physically separated from the invoking program, either by residing in separate files or by being placed after the logical end of the invoking program. A significant advantage of external subprograms is that the scope of variables is the opposite of that for internal ones: they are local by default unless declared global. This allows finer control over intended and inadvertent side effects.

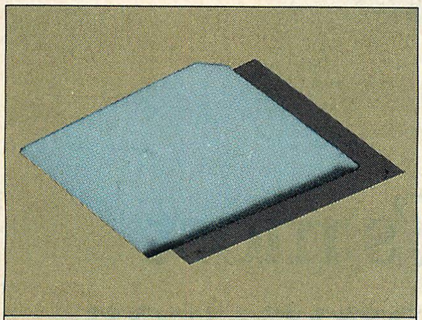
WATCOM BASIC supports internal subprograms only. Except for the primitive GOSUB procedures and single-line functions, BetterBASIC and BB^x support only external programs. True BASIC and MEGABASIC support both. In addition, BetterBASIC, True BASIC, and MEGABASIC all implement libraries of precompiled external routines.

FIELD statement. BASICA's FIELD statement is often cited as one of its worst features. Its purpose is to specify the fields that make up a record of a random-access file. Record-oriented file I/O is not usually supported in BASIC to the extent that it is in COBOL or Pascal. BASIC has no standardized way of specifying the layout of a file record, and the implementations in the various dialects of the language are significantly different. Alternative methods of specifying record layouts are not necessarily any better or easier to code than the much-maligned FIELD statement.

The big shortcoming of the FIELD statement is that each of the fields in the record can contain only string, not numeric, data. In order to write numeric data to a random file, the system must be fooled into treating it as string data. This is accomplished by means of the MKI$, MKS$, and MKD$ functions on output and the CVI, CVS, and CVD functions on input. This scheme negates one of the advantages of record I/O: the aggregation of different data types into a single structure. Apart from this drawback, the FIELD statement is not too bad; each field is identified by name, and although fixed record length is restrictive, it lessens the potential for programmer errors.

Because of their goals of compatibility with BASICA, both Professional BASIC and BetterBASIC implement the FIELD statement in the same way. BetterBASIC also offers an alternative means of record I/O that is by far the best of all the BASICs reviewed here. It implements a STRUCTURE data type, which is described below.

The record I/O of the other products makes BASICA's FIELD statement appear quite effective in comparison. Most of them require constructing a record by string concatenation, but provide no means of guaranteeing the correct record length.



BetterBASIC

At first glance, Summit Software's BetterBASIC seems to be a no-lose proposition. The current release (2.0) is almost fully compatible with BASICA, yet it provides many extensions that make it suitable for serious applications development. It can even read BASICA tokenized or ASCII files directly.

BetterBASIC is halfway between a compiler and an interpreter. Program lines are checked for syntax at entry, and converted to an intermediate code that is processed much further than BASICA's tokenized format. This makes screen response in the editor somewhat sluggish, because the program must be recompiled for each new or changed line, but it results in a quicker execution time and the location of all errors before execution begins. Otherwise, the screen interface is quite similar to BASICA's. An optional runtime system allows the creation of stand-alone .EXE files that are executable from DOS.

Foremost among BetterBASIC's extensions is full utilization of memory: more than 300KB of program and data space on a 640KB system. Furthermore, a full set of structured programming constructs is provided: multiline IFs, several forms of FOR, DO, and WHILE statements (but no CASE), and call-by-name subroutines and functions with passed parameters and local variables. GOTOs and GOSUBs are supported for compatibility with BASICA; however, structuring is enforced by disallowing branches into a loop and requiring only one NEXT per FOR statement.

BetterBASIC strings and arrays may be either static (fixed size) or dynamic (size may change during execution). The manual recommends the former for efficiency, but the more flexible dynamic storage allocation is available when needed. Two advanced features are borrowed from the C language: structures and pointers. A structure is a collection of various data types into a single named entity. It is especially useful for specifying the layout of file records and is easier to use than FIELD (also available in BetterBASIC). Because of structures, BetterBASIC's handling of record-oriented file I/O is the best of all the products reviewed.

The STRUCTURE statement, similar in concept to Pascal's RECORD, is a list of data fields of various types and lengths collected into one named entity. The big difference between STRUCTURE and FIELD is that fields are not limited to string data, but may be declared as any type supported by the language. For the numeric and pointer types, the type itself sets the length; for strings, the length is specified in the record definition. (Strings within records are static; their length cannot change dynamically.) Unlike the FIELD statement, which is defined for a particular file, a record name has no relation to a file until it is mentioned in an I/O statement. The fields of a structure may be used just as any other BASIC variables, without the need for special conversion (MKI$, CVI, etc.) or assignment functions (LSET, RSET).

Pointers can be used for implementing linked lists in memory or for saving the address of an oft-referenced element of a multidimensional array (this saves the time of repeatedly recalculating the address). Unlike C, however, BetterBASIC does not allow manipulating the value of the pointer itself, only the item pointed to. For example, if *X* is a pointer to an integer, *X* + 1 increments the value of the integer that *X* points to; it does not increment *X* to point to the next integer in an array or structure. Pointers are manipulated with the SET statement, and they may be set to point to a variable or to wherever another pointer is pointing. Therefore, if pointer *X* points to *A*(*I*), it is moved to the next element of *A* by the statement SET X TO A(I+1). If *X* and *Y* are pointers to the same type of data, then the statement SET X TO Y will set *X* to point to the same data that *Y* is pointing to (that is how a program would the pointers in a list structure).

Except for the primitive GOSUB procedures and single-line functions, BetterBASIC supports only external subprograms. The implementation is similar to that in advanced structured languages such as C and Pascal: arguments are passed either by value or by reference, and variables are local unless declared public. Unlike the other BASICs, procedures are invoked simply by using their names, not by a CALL statement. Furthermore, arguments need not be enclosed in parentheses, so the name of a procedure is used as if it were a key word of the language.

The concept of families is an interesting feature of BetterBASIC's implementation of subprograms. A number of subprograms (procedures or functions) may be defined with the same name but different argument lists. When a subprogram of that name is invoked, the system matches the argument list with the declared parameter list of each member of the family and executes the one whose parameters match the actual arguments in number and type. Another interesting feature is that arguments may be declared optional; if omitted, they take on a default value. Further, BASICA has a key-word argument type that is somewhat analogous to enumerated data types of Pascal, allowing the assignment of descriptive names to each of a range of integer values.

One criticism about BetterBASIC subprograms is the way that library files are handled. External programs may be appended to the calling program or collected in library modules. In the latter case, however, the library files needed by a given program are named not in the calling program, but in a configuration file that is processed when BetterBASIC is loaded. This makes it quite inconvenient to tailor the list of libraries for each application.

Because of structures, BetterBASIC's handling of record-oriented file I/O is the best of all the enhanced BASICs reviewed here.

Its handling of variable declarations is perhaps BetterBASIC's worst feature. As in Pascal and C, the type of each

How to make IBM's® most powerful personal computer even more powerful.

PLUG IN QUADBOARD®-AT

Quadboard® a sophisticated combination of multifunctions and expanded memory. Quadboard-AT unlocks the full potential of the PC AT, to make IBM's most powerful personal computer even more powerful.

POWERFUL MULTI-FUNCTIONS WITH EXPANDED MEMORY.

Quadboard-AT begins with a printer port and two communications ports. Add on printers, modems, or other communications devices in a snap. **And because Quadboard-AT features expanded memory, you can take full advantage of the AT's™ powerful RAM capabilities.** Add from 512K to 3.5 megabytes of extra memory to process information faster and more efficiently than ever before.

Fully Compatible with
EMS
Expanded Memory Specification

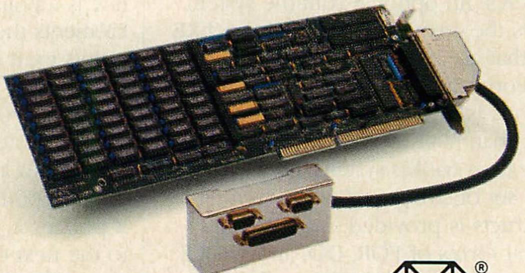
COMPATIBILITY FOR THE FUTURE.

With Quadboard-AT, you get **full compatibility with the expanded versions of all your favorite software.** Like Lotus 1-2-3™, Symphony™, and Windows®. While other boards lock you in to the 640K DOS limit, Quadboard-AT gives you freedom to grow. Now and in the future.

THE BOTTOM LINE IS VALUE.

Quadboard-AT comes with QuadMaster® III utility software and a two year warranty (1.5 Mb versions). No other AT board on the market delivers more for the money. Visit the Quadram® dealer nearest you to get the full story on this powerful new enhancement. Or contact us at One Quad Way, Norcross, Georgia 30093-2919; 404-564-5566.

QUADBOARD-AT	AST ADVANTAGE!™
☑ EXPANDED MEMORY. Fully compatible.	☐ EXPANDED MEMORY. Incompatible.
☑ FEATURES. Two serial ports standard.	☐ FEATURES. One serial port standard.
☑ RELIABILITY. Two year warranty.	☐ RELIABILITY. One year warranty.
☑ VALUE. The choice is obvious.	☐ VALUE Less for the money.



QUADRAM®
An Intelligent Systems Company

CIRCLE NO. 147 ON READER SERVICE CARD

variable (BYTE, INTEGER, REAL, STRING, POINTER, STRUCTURE, or an array of any of these) must be declared at the beginning of the program. Although this is at odds with the spirit of the BASIC language, in itself it is not as onerous as it may seem. An AUTODEF switch can be set to generate declarations automatically from the first use of each variable in the program.

The problem is that BetterBASIC hides declarations from the programmer. Type declarations are difficult to display and sometimes impossible to change. Whether typed by the user or generated by AUTODEF, declarations are entered without line numbers as if they were immediate commands, and they are not displayed with the program by the LIST command. Declarations can be displayed only by LIST DATA, or both declarations and the entire program can be displayed by LIST ALL.

Even worse, these declarations are very difficult to change from within BetterBASIC. Attempts to redeclare a variable with a different type or to change the size of an array generate the message "Variable already defined." The CHANGE command alters the type of a numeric variable or the size of a static string, but the program does not allow the user to change the size of a static array, switch a variable from array to a scalar, or delete an array declaration.

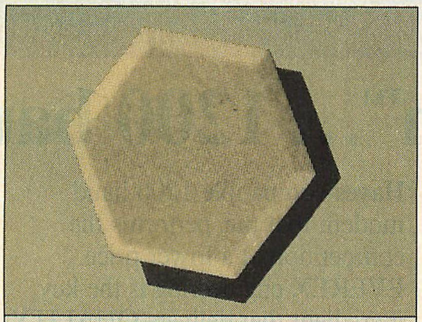
Working around the declaration drawbacks is possible (though inconvenient). The DELETE command removes a scalar, but not an array, from the declaration section. An array declaration can be deleted if it is in the BASICA DIM format, because BetterBASIC is sufficiently compatible to allow the same syntax. When all else fails, declarations may be changed by writing out the program with its declarations to an ASCII file, exiting BetterBASIC, modifying the file with a text editor, then reading it back into BetterBASIC. The declaration scheme is one of the roughest of several rough edges in BetterBASIC.

The screen editor is essentially the same as BASICA's, but as previously mentioned, its response is sluggish enough to negate much of the immediacy of an interactive interpreter. The Enter key generates keystrokes faster than the interpreter can process each line. Pressing Enter on a previously displayed OK prompt generates an error message (in BASICA, the OK line is blanked out with no message). Reading an ASCII file often generates an "Input past end" message, even though the file is read and interpreted correctly. If an ASCII source file contains errors, those lines are listed and flagged but do not become part of the program in memory; converting them to comments would be more useful.

Compatibility with BASICA is good but not complete. All BASICA statements are implemented except the USR function (an alternative way to call assembly language functions is provided) and those that provide support for cassette I/O, light pen, and game adapter. Some statements are partially implemented. In particular, DRAW supports only a subset of the BASICA graphics language: values in the DRAW string must be constants, not variables, and execution of a substring with the X subcommand is not supported. These are serious limitations of the BASICA graphics capabilities.

Documentation is provided in a thick binder. It is not typeset, but the text is easily readable. The early sections are geared to the novice, whereas the reference section is short on explanations and examples, especially for the features that BetterBASIC shares with BASICA. This manual should be used along with, not instead of, BASICA's. Editing is a bit careless; some of the key words are out of alphabetic sequence, and some explanations have not been fully updated to reflect version 2.0.

The underlying design philosophy of BetterBASIC is quite good, and it has several impressive features. The current implementation, however, still leaves a little to be desired.



Business BASIC

The noteworthy features of BB^x are binary-coded decimal (BCD) arithmetic with 14-digit accuracy, call-by-name subroutines with local arguments, and implementation of keyed files. This system, which was ported to DOS from UNIX, makes no concessions to BASICA compatibility or even to any of the conveniences provided by DOS. The language is very limited, with no support for communications, graphics, sound, or event trapping. Even PEEK and POKE, which are staples of most BASIC dialects, are not implemented.

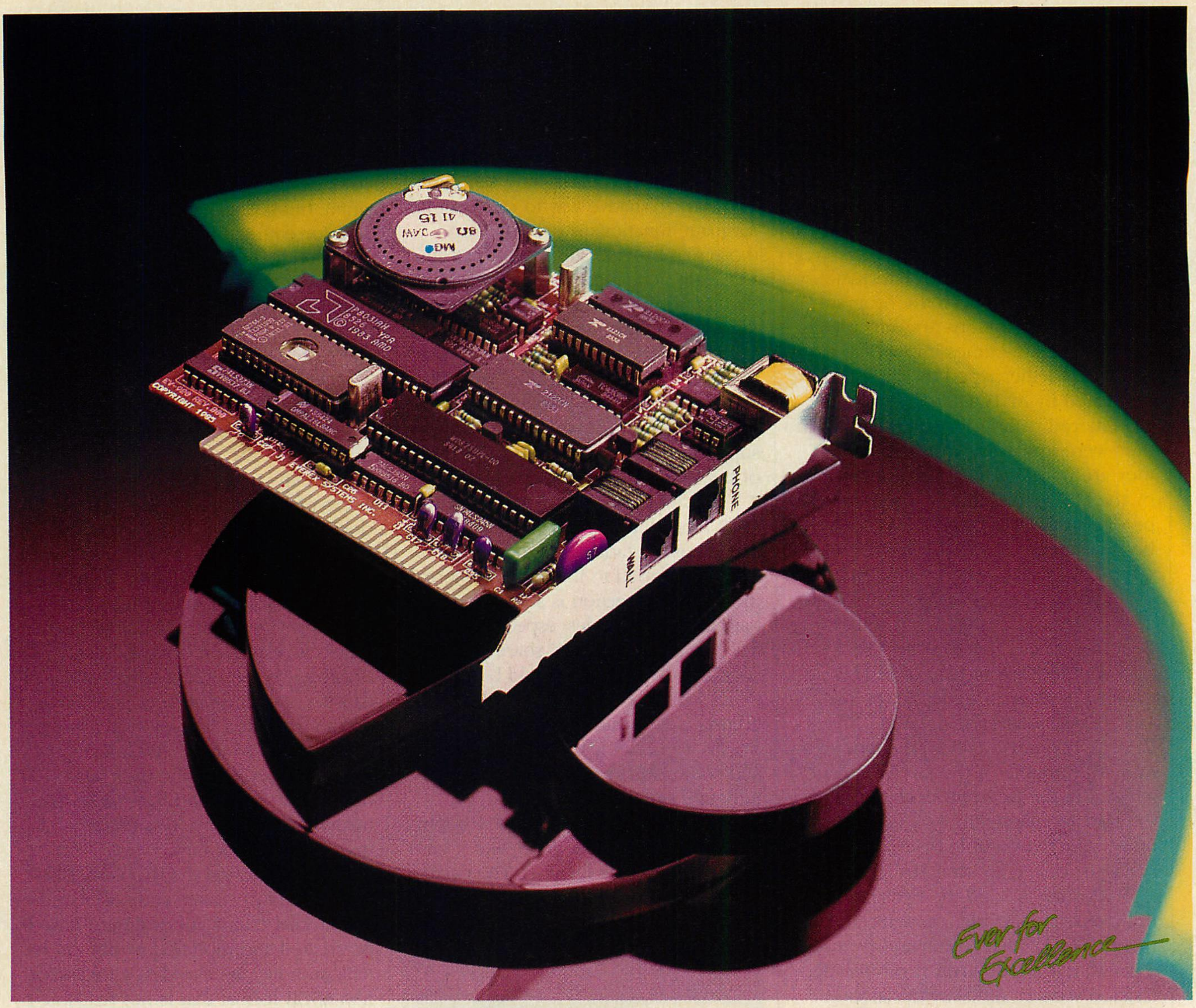
The BB^x system lacks a functional program editor. Full-screen cursor control is not provided. The only available editing key is the destructive backspace, and that is in effect only while a line is being entered. An EDIT command allows changing existing lines in the program, but it does not present the target line for editing by overtyping, inserting, or deleting. Instead, the user must specify how much of the line to retain, how much to delete, and where to replace or insert text. In most cases, retyping the whole line is much easier.

A utility program, written in BB^x itself, purports to be a full-screen editor for BB^x programs. It is activated by an immediate CALL command and allows full-screen cursor control as well as overtype, insert, and delete editing. The documentation for the edit program is poor, and it advises the user to "experiment with the edit program as an alternative to the EDIT command." A help file is provided, but it is not accessible unless it resides on drive A:. A serious drawback is the lack of large-scale navigation commands, such as page down, go to a specified line, or search for a given string. To move 100 lines up or down in a program, the user must lean on the cursor key and wait while that many lines scroll by. This editor is of marginal utility at best.

To compound the problem, BB^x cannot read in source code prepared by an external editor. All programs have to be created and modified within the system, and they are saved in a tokenized, non-ASCII format that is not readable by any other program.

When saving a program for the first time, the size of the file must be specified along with its name. Fortunately, if the size is too small for the program, it is extended, but if too large, disk space is wasted. Random and keyed files must have a record count specified at creation; they cannot be extended. Before a program of any size is run, sufficient memory must be allocated.

The use of decimal floating-point representation for all numbers (integers are not implemented) takes its toll in memory usage, because each of these numbers is ten bytes long. Two versions of BB^x are supplied: a small model with 32KB of workspace and a large 64KB model. As a result, BB^x could not run the standard sieve benchmark with an array dimensioned to 8,191, because that would require more than 80KB of memory. Although all arithmetic is performed in decimal, functions are included to convert values to and from strings containing binary



Evercom™... 1200 baud, $249

Now the EVEREX Evercom modem is available on a half-size card, perfect for the short slots in the IBM XT, Portable, and many compatibles. EVEREX has combined innovative engineering and technology to produce a quality Hayes-compatible 1200 baud modem that out performs the competition at a lower price. EVEREX engineering is the key ingredient that delivers QUALITY and FEATURES without sacrificing PRICE.

Before you buy a modem, check these features:

	EVERCOM	HAYES 1200B
• 300/1200 bps Hayes-compatible	YES	YES
• Half-size card	YES	YES
• Automatic dialing, answering, and redialing	YES	YES
• Call Progress Monitoring	YES	YES
• Configurable from COM1 - COM4	YES	NO
• Internal Speaker with software adjustable volume control	YES	YES
• Tone and pulse dialing	YES	YES
• Automatic data-to-voice transition	YES	NO
• Detects receiver off-hook	YES	NO
• Reports speed mismatch	YES	NO
• Supports 132 columns	YES	NO
• Communications software included	YES	NO
• Extended Hayes command set	YES	–
• **List price**	**$249**	**$489**

Demonstration -- FREE Brochure

Remember, before you buy a modem, check the features, check the price, and then call EVEREX to set up a demonstration at your local dealer.

When you call, we will also send you the total solution brochure describing EVEREX's complete line of IBM microcomputer peripherals.

1-800-821-0806

in California

1-800-821-0807

EVEREX

EVER for EXcellence

48431 Milmont Dr. Fremont CA 94539
(415) 498-1111

CIRCLE NO. 134 ON READER SERVICE CARD

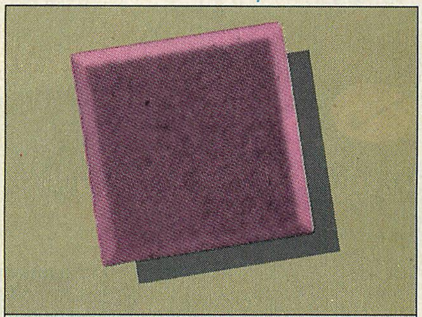
IEEE format. Such values, when written to data files, may be read by programs written in other languages.

The file system supports three types of files: byte, random, and keyed. Random and keyed files have fixed-length records up to 32KB each; keys can be up to 64 bytes. A record in a keyed file can be found simply by specifying a key value in the READ or WRITE statement. The file system has two problems. First, text files with variable-length records delimited by CR/LF are not supported in a useful way. Although scanning file input and branching out of the I/O statement on specified input characters (such as CR and LF) is possible in a convoluted way, it is no match for the convenience of BASICA's LINE INPUT statement.

Second (and this is ironic for a product with *Business* as part of its name), BB^x has no concept of a file record as a collection of fields of various data types. Each record is read into or written from a single string variable, but no special functions are provided to guarantee that the concatenation of fields is of the correct length. While numeric variables can be read or written directly with standard list I/O, true record I/O is not accomplished so easily. The documentation is not clear on this point, but it appears that the only way to write numeric data to a record file is to convert the data to ASCII or IEEE strings and then concatenate them into a single string the length of one record. The former exacts a penalty in both processing time and storage space, while the latter loses accuracy, which is the reason for using decimal representation in the first place.

The BB^x documentation suffers from a lack of detail. The manual is primarily a reference with very little explanation and even fewer examples. One of the program's most useful features, keyed files, is inadequately covered. Although two kinds of keyed files are mentioned—sort files containing keys only and direct files containing both keys and data—the use of sort files is not explained. Are they to be used when the whole record is the key or for pointing to data records in another file? These questions are not answered.

One bright spot in BB^x is the implementation of external procedures (external functions are not implemented). Call parameters may be passed either by value or by reference. Variables may be local or global, but this is an all-or-nothing proposition: either all the variables of the caller or none of them except the parameters are accessible from the called program. The program has no capability to share only some of the variables. With local variables, the procedure may call itself recursively. An external program resides in a separate file and is loaded from disk at the time it is called; it is normally dropped from memory when it exits. However, frequently called procedures can be made resident for the duration of the main program's execution. This feature works well and is adequately documented, but it does not make up for the shortcomings of BB^x.



MEGABASIC

American Planning Corporation's product, known as APC BASIC in earlier incarnations, is a well-conceived implementation of BASIC, but it is at a disadvantage on the PC because it is written for a generic DOS machine. That means it offers no graphics, communications, or sound support, because each of these features must be tailored to a specific hardware configuration. Apart from this lack of full support for the PC hardware, MEGABASIC's only other major disappointment is the lack of a multiline IF structure. Extensions to BASICA's capabilities include full memory utilization, loading of several programs into separate workspaces, recursive subprogram calls, and low-level system access. MEGABASIC can even be used for writing interrupt service procedures.

MEGABASIC is a true interpreter. Like BASICA, it does not discover most errors until the flow of execution reaches them. Once programs are fully developed, they may be executed by the supplied runtime system, which is merely the interpreter without the editor and immediate command processor. It runs slightly faster than the full system and produces tokenized source files that are somewhat smaller.

MEGABASIC's internal editor is not very good, as might be expected from a generic implementation. The EDIT command enters a mode similar to the update mode of EDLIN, but the edit mode is controlled by Ctrl-letter combinations instead of function keys. Block move and search and replace functions are provided, and the editor keeps track of unsaved changes and asks for permission to delete them when loading a new file or exiting from BASIC.

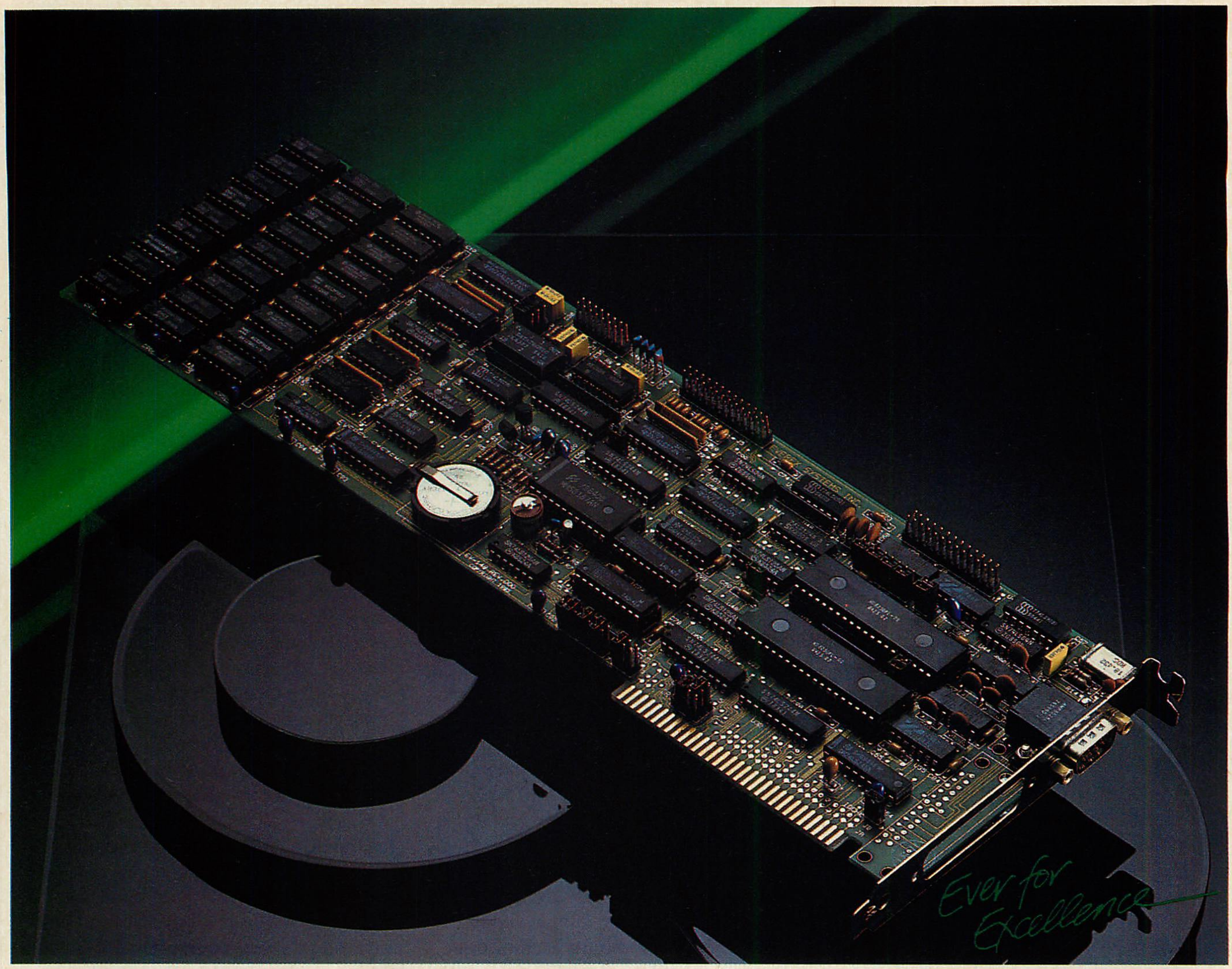
Source code may be read and saved in either ASCII or tokenized format, so an external editor can be used. Saving ASCII files is not very convenient, because the program must be written to a file number. The file is first opened to assign a name to the number, then the program is saved, and the file must be closed. If left open, the next save appends another copy of the program to the end of the file.

MEGABASIC integers are two words long, able to represent numbers with an absolute value of more than 2 billion. Real numbers are represented in decimal floating-point format, with the accuracy selectable between 8 and 14 digits. Variable names may be as many as 250 characters long, and all characters are significant. That limit is obviously not practical because each program line is limited to 255 characters. Strings are static, meaning that their maximum length must be declared. In effect, a string is an array of characters and always consumes the same amount of memory. As in BASICA, undimensioned arrays default to a size of 10.

String handling is definitely MEGABASIC's strong suit. It has a comprehensive set of string operations not found in most other BASICs. Besides the common concatenation and comparison, MEGABASIC provides operators for taking the difference, maximum, minimum, and match of two strings; other operators perform bitwise logical operations (AND, OR, NOT, XOR, etc.).

The DIFFERENCE operator is interesting: the result of A$-B$ is a string with only the characters that appear in A$ but not B$. One typical use would be count occurrences of particular characters in a string. LEN(A$)–LEN(A$–"x") determines how many times *x* appears in A$. The MATCH operator builds a string with CHR$(255) in every position where two strings are equal, and CHR$(0) where not equal. MIN and MAX get the lower or higher of the characters in corresponding positions. MEGABASIC has many other string functions, including comprehensive parsing and searching capabilities.

This implementation of the language suffers some annoying incompatibilities with BASICA. In some cases the syntax seems to have been arbitrarily changed: multiple statements on one line are separated with semicolons, not



Magic Card II ...Multifunction, $199

The EVEREX Magic Card II is the multifunction card that has the features you want at a price you can afford. It can even take an original PC, with a 64K motherboard, to 640K of RAM. Installation is a snap with our unique TESTER program that runs independently and shows exactly where to install RAM and how to set the options.

Why pay almost twice as much to get less? EVEREX combines innovative engineering, technology, and quality manufacturing to bring you a quality product that out performs the competition at a lower price. EVEREX engineering is the key ingredient that delivers QUALITY and FEATURES without sacrificing PRICE.

Before you buy a multifunction card, check these features:

	EVEREX Magic Card II	AST Sixpak Plus
• Uses 64K or 256K Ram Chips	YES ✓	NO
• Battery Backed Clock/Calender	YES	YES
• Parallel Port Standard- LPT1—LPT3	YES	YES
• Two Serial Ports Standard- COM1—COM4	YES ✓	NO
• Game Port Standard	YES ✓	NO
• Adds up to 576K RAM	YES ✓	NO
• RAM disk and print spooler software	YES	YES
• Available with zero RAM	YES ✓	NO
• Compatible with 8MHZ operation	YES ✓	NO
• **List Price**	$199 -0K	$395 -64K

Magic Card II in Action

See for yourself what the Magic Card II can do. Call EVEREX today for the name of the EVEREX dealer near you.

You'll be convinced that—EVEREX is EVER for EXcellence too!

1-800-821-0806

in California

1-800-821-0807

EVEREX

EVER for EXcellence

48431 Milmont Dr. Fremont CA 94539

(415) 498-1111

Magic Card II is a trademark of EVEREX SYSTEMS, Inc.

SixPak Plus is a registered trademark of AST Research, Inc.

CIRCLE NO. 135 ON READER SERVICE CARD

colons; the termination of a WHILE loop is NEXT, not WEND; no PEEK and POKE functions are included, only similar functions that read or write to memory as if to a file; segments are defined with SEG instead of DEF SEG.

MEGABASIC's file system supports text and binary files. The implementation of text files holds no surprises. Binary files, however, are unusual in that they have no concept of a record; the file is considered a string of bytes and positioning is by byte count. To read a particular record requires calculating a byte offset by multiplying the record number by the record length. The layout of the record is specified in each I/O statement by giving the length and type of data to be read into each variable in the I/O list. Whether this is an improvement over BASICA's FIELD statement is questionable. The cryptic length and type codes interspersed among the variables in the I/O list make for difficult reading. Furthermore, program maintenance is made more difficult by hiding the record layout in the executable code and replicating it in multiple I/O statements instead of isolating it in a declaration section at the beginning of the program.

Of all the products reviewed here, MEGABASIC provides the most sophisticated facilities for modular programming. It implements recursive internal procedures and functions with arguments passed either by value or by reference. All variables are global unless expressly declared local.

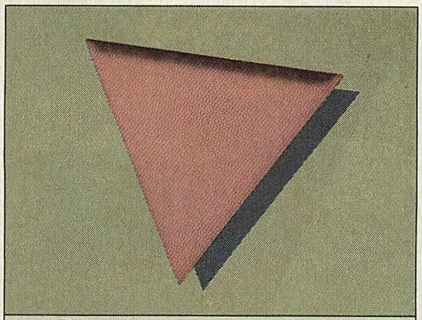
MEGABASIC also supports external program modules in separate files. Up to 64 of these may be loaded into memory at once, each into its own workspace (provided that the system has sufficient memory). By default, the scope of subprogram names and variables is local to that module, but each module may declare public resources (functions, procedures, and data variables) that are to be accessible from other modules. At runtime, modules are loaded into memory only when specifically requested and remain there until specifically dropped. A module may have a prologue procedure that is executed at load time and an epilogue that runs when the module is dropped.

Once a module is loaded into memory, its public resources are not accessible to other modules until the scope of accessibility is defined. A module can allow other modules to access its public resources, request access to the public resources of other modules, or specify that access be allowed from any module to any or all other modules. This provides a great deal of control over the scope of variables and other program resources. It also facilitates the implemention of large-scale modular applications.

The interface to assembly language routines is somewhat unusual in MEGABASIC, because parameters are passed only via the registers, not on the stack. The user accesses the machine registers from BASIC by concatenating the value for each register into a string and calling the assembly language interface routine that actually loads them.

MEGABASIC's documentation contains just the right blend of reference material and explanation. Organization would be improved if each major section presented commands, key words, functions, etc. in alphabetic order rather than by subject. The text is typewritten and reduced in reproduction, so it is a little hard to read.

MEGABASIC is quite impressive for what it is: a BASIC interpreter for a plain-Jane DOS machine. Lack of PC-specific features, however, significantly reduces its usefulness.



PROFESSIONAL BASIC

Unlike the other products reviewed, which are dialects of the BASIC language and/or supersets of BASICA, Professional BASIC is a development environment for BASICA programs. Specifically, it provides comprehensive debugging facilities to trace the execution of BASICA programs, using an involved windowing system. (See photo 1 for a typical split-screen window display.) Most of the BASICA capabilities are implemented, but with significant omissions. Specifically, these major commands are not available: BLOAD and BSAVE, DRAW, graphics GET and PUT, PLAY, VIEW, and WINDOW. Nor is support provided for interrupt-driven serial communications, light pen, and game port. If these facilities are not essential, Professional BASIC can be a useful tool.

The design philosophy underlying Professional BASIC is to increase communications between program and programmer over that provided solely by the program's output. To this end, the system provides 14 different windows into various aspects of the program's operation. In some respects, this is an embarrassment of riches, and the user is easily lost in the window structure.

Professional BASIC reads and writes only ASCII source files. When a program is loaded from disk or typed from the keyboard, each source line is semicompiled into an intermediate pseudocode, which cannot be saved. All source errors are found before execution begins. The editor is very similar to BASICA's—although different enough to be annoying. The cursor cannot be moved over the screen at will; instead, the F9 and F10 keys copy the previous or next source line onto the command line at the bottom of the screen; there it may be edited as if by the EDIT command. Alternatively, the user can enter the EDIT command with a line number, as is the case in BASICA.

The Professional BASIC editor provides several enhancements, including a SEARCH command that lists all lines containing a given string or highlights all occurrences of that string in the statements displayed on screen. The value of this feature is compromised by the lack of full-screen cursor control. A SORT command lists all variables in alphabetic sequence, but does not create a cross-reference of line numbers where the variables are used.

Professional BASIC has two basic types of windows: execution and tracing. When the program is loaded, it brings up the execution system's control window. There, the user loads the program, performs editing, and issues immediate commands. Another execution-type window is the print window, which appears when the RUN command is issued and which receives the program's PRINT output and echoes its keyboard input. The control window is reinstated when the program terminates or is interrupted with Break.

The tracing system, which contains 12 windows, is entered by invoking the SRUN command or by pressing an Alt-letter combination that identifies the window. Once within the trace system, single-letter commands switch between the various windows, including the execution windows, or display any two trace windows side-by-side. The trace windows are listed below.

- The list window is entered by the SRUN command or by execution of a breakpoint in the program. It shows a structured listing of the program with single statements per line, multiline

Gateway to anyware.

Thanks to our local and remote communications solutions, your PCs now have total access to the world around them. They can talk to other PCs. Share hardware and software. And take advantage of all the major networking standards. From NetWare to NETBIOS to SNA to X.25.

G/NET™ is our high-performance, low-cost Local Area Network for IBM PCs, XTs, ATs and compatibles. Incorporating the Novell NetWare/G family of LAN operating systems, G/NET offers electronic mail, print spooling, multiple file and communications servers, and access to over 3,000 LAN applications.

And with our G/NETBIOS™ emulation option, G/NET can even run programs developed for IBM's PC Network or Token Ring. This gives you the best of both worlds: IBM LAN software compatibility coupled with the performance advantages of G/NET hardware.

Our G/SNAnet™ gateway provides SNA mainframe links for networked PCs. Up to 32 concurrent sessions are supported through one shared, board-level controller. Each PC can switch between 3270 emulation, 3770/RJE emulation, and local PC-DOS or LAN applications.

Our G/X25net™ LAN gateway lets multiple G/NET users access remote LANs, mainframes and dial-up services. Over X.25 trunk lines or public networks such as Telenet and Tymnet. Finally, our G/Async™ and G/Carbon Copy™ links can be used in a variety of async applications.

Think of us as your PC SafetyNet.™ In addition to our cost-effective products, we offer the security of a total solution supplier. With complete service and support. Continued innovation so you won't be left behind with yesterday's technology. And assured performance and reliability. Just ask any one of our more than 50,000 users.

So call today for your local Gateway distributor or dealer. And let your PCs explore our gateway to anyware.

Gateway communications, inc.

Your PC SafetyNet.™

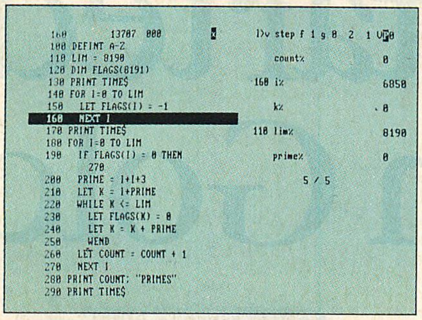
CIRCLE NO. 141 ON READER SERVICE CARD

16782 Red Hill Ave., Irvine, CA 92714. (714) 261-0762.

IFs, and indented block structure for IFs and loops. As the program is executed in single-step or sequential mode, each statement is highlighted in inverse video. Pressing Enter begins execution; pressing the Space Bar suspends execution or executes only the next statement. Options within the list window allow statements that have not yet been executed to be highlighted; the user also can display counts and histograms of execution frequency for each statement. On a system with a color/graphics adapter, the highlighting is in blue; on a monochrome monitor, however, those lines would be unreadable. A FINETRACE option shows the progress through the component parts of a statement—for example, a complicated arithmetic expression.

- The time window displays program statements in the order they are executed. When tracing a loop, the loop statements are listed anew for each iteration. (In the list window, the highlight returns to the top of the loop on each iteration; the loop statements appear only once.) The value of this window is questionable, because the list window shows the same information in a more useful fashion.
- The pseudocode window traces execution through the elementary pseudocode instructions that are generated from the source code. The code is displayed in understandable English; for example, the statement X% = Y% + Z% generates "Load value of Y% to IA, Add value of C% to IA, Store IA to A%." IA is one of two integer registers; also provided are two single-precision, two double-precision, two string, and three index registers. At the user's option, the applicable registers can be displayed at the top of the window. For most programs, the pseudocode window has marginal value, because it applies more to the internal operation of Professional BASIC than to the operation of the program being traced.
- Five windows are available for tracing data values. The data window shows a list of DATA statements and highlights the next value to be read. The variable window displays the values of all scalars in the program, the array window shows arrays in a single column, and the matrix window has two-dimensional, single-precision arrays in rows and columns. Navigation keys are used to scroll to the beginning of the next array or to scroll by as many as 10,000 elements in either direction. The fifth window is for disk I/O and

PHOTO 1: *Trace Windows*



Professional BASIC's trace windows show indented code of SIEVE.BAS on the left, array updates on the right.

shows the contents of file buffers for each open file.

- The memory window displays the contents of an arbitrary memory address in the segment specified in the last DEF SEG statement; it is especially useful for tracing PEEKs and POKEs.
- The FOR/NEXT window displays up to four active FOR loops, showing the loop limits and the current value of the loop variable.
- The GOSUB window shows the path through nested subprogram calls.
- Finally, the print trace window is a copy of the execution system's print window. This allows splitting the screen between print output and any of the trace windows (the screen cannot be split in the execution system). The two windows can be automatically switched each time the output goes to the side of the screen containing the trace window.

As mentioned previously, Professional BASIC is intended primarily for developing programs, not running applications. The manual suggests that applications be designed to run under BASICA or even be compiled with a BASIC compiler.

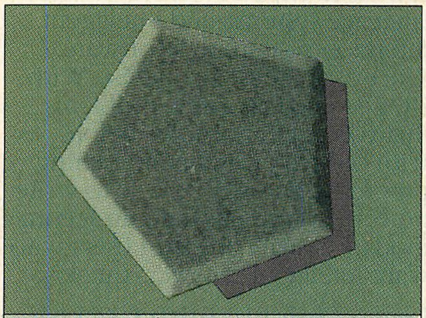
Other details underscore the fact that Professional BASIC is a developer's tool rather than a vehicle for end-user applications. BASICA lets the developer protect an application with the ,P option of the SAVE command. This option encodes a source file and prevents an end user from listing or editing it. Some developers use ,P for modular programming, protecting each section of code after it has been thoroughly tested. Professional BASIC has no equivalent of the ,P command; source is always displayed and the editor is always active.

Some enhancements to Professional BASIC preclude its use if BASICA compatibility is important. Foremost among these is the use of named labels instead of line numbers as targets of GOTOs and GOSUBs. Program lines can be up to 311 characters long; variable names can have 308 significant characters; and integers are two words long, allowing absolute values of more than 2 billion. However, integer data written to files via a FIELD statement can be limited to two bytes, if desired.

Other major differences between Professional BASIC and BASICA are that all arrays must be dimensioned (they do not default to 10 elements), arrays and scalars must have distinct names, and only one NEXT per FOR and one WEND per WHILE are allowed. These requirements, which are more restrictive (but desirable from the standpoint of programming style) in Professional BASIC than in BASICA, might prevent a BASICA program from running in Professional BASIC, but not vice versa.

The Professional BASIC manual does not claim to be an exhaustive reference; the user is warned to use it as a supplement, not a replacement, for the BASICA documentation. The explanation of the labyrinthine window system is not very well organized and contains no reference section. Instead, the user is walked through many example programs (all provided on disk). This is fine for learning, but not very useful several months down the road in the middle of a debugging session.

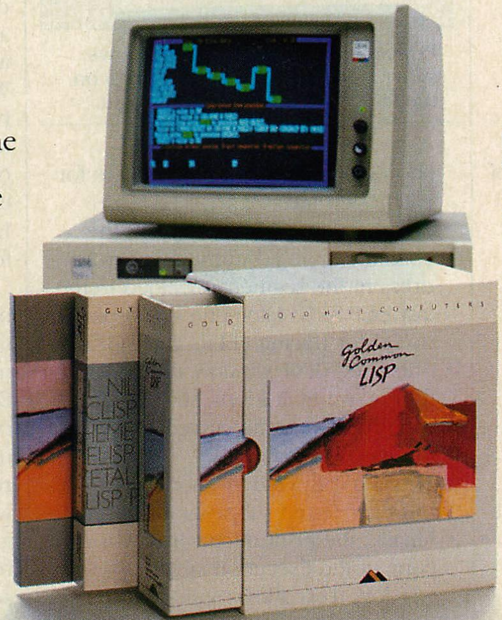
Professional BASIC is a novel concept, well executed, and could be very useful for developing applications that do not need the missing features.



TRUE BASIC

Developed by the inventors of the original BASIC language, True BASIC boasts a few advantages over BASICA—the primary one being the use of structured programming constructs, such as multiline IF, DO WHILE, DO UNTIL, and CASE statements and separately compiled subprograms with local variables. Older style GOTO and GOSUB statements are also supported for compatibility with earlier versions of the language. Line numbers are optional, but

Get started in AI with Gold Hill.



GOLDEN COMMON LISP is the best selling AI language.

You know artificial intelligence is the wave of the future. Programs based on the ideas of artificial intelligence are being written today in COMMON LISP–the new LISP standard developed by researchers from universities and corporations such as Carnegie Mellon, DEC, MIT, Stanford, Symbolics and TI. LISP allows the development of programs that are intelligent, flexible, and even human-like. The problem has been that hardware needed to run LISP is expensive.

No longer. GOLDEN COMMON LISP (GC LISP) makes it possible for you to learn and use LISP on your *personal computer.* When you use GC LISP in combination with the AI examples provided, you will be able to develop such exciting applications as expert systems, natural language systems, and intelligent interfaces to complex software. The possibilities are endless. With GC LISP you can tackle problems that you could not solve before with ordinary number crunching programs. You will be able to write powerful programs that can accommodate casual computer users who want intelligible answers quickly.

Learn Quickly with GC LISP

GC LISP makes it easy for you to learn the technology of AI at your own pace. Each package includes the *San Marco LISP Explorer®*–an interactive software tutorial developed by San Marco Associates with Patrick H. Winston. This 800-frame instructional system guides you through the steps of LISP programming and makes the full range of LISP's power accessible to both novice and experienced programmers. The second edition of the classic *LISP* textbook by Winston and Horn is also included.

GC LISP comes complete with the intelligent GMACS editor (based on EMACS), full on-line documentation of all GC LISP and GMACS functions, the *Common LISP Reference Manual* by Steele, and a comprehensive user manual. In short, GOLDEN COMMON LISP comes with everything you need to program in LISP.

NEW! The GC LISP 286 DEVELOPER

Discover programming power you never thought possible on your PC AT! GC LISP 286 DEVELOPER is a complete programming environment designed for GC LISP-based applications requiring extended memory and faster execution. The 286 DEVELOPER includes a Large Memory Interpreter capable of addressing 15 megabytes of physical memory, a Large Memory Compiler that runs applications up to 15 times faster, and a complete set of program development features including lexical and dynamic scoping. The 286 DEVELOPER is the first PC-DOS programming environment that takes full advantage of the speed and memory capacity of the IBM PC AT and compatibles, and it's now available to help you develop powerful and productive AI applications.

Order Today

For more information on Gold Hill products, please call today:

1-800-242-LISP

In Massachusetts call collect (617) 492-2071

GC LISP requires an IBM PC, PC XT, PC AT, or IBM-compatible with a minimum of 512K and PC-DOS 2.0 or higher. GC LISP 286 DEVELOPER is non-copy protected, and requires an IBM PC AT or compatible with a minimum of 512K of base memory, 2 megabytes of extended memory, and PC-DOS 3.0 or higher.

GOLD HILL COMPUTERS

163 Harvard Street, Cambridge, Massachusetts 02139

CIRCLE NO. 122 ON READER SERVICE CARD

they are an all-or-nothing proposition: either every line is numbered or none is. Numbering is required when old-style branch statements are used.

One feature of True BASIC that has received a lot of criticism is the fact that assignment statements must begin with the LET key word. This is a requirement of the proposed ANSI standard for the BASIC language (see sidebar).

True BASIC's user interface harks back to the time-sharing heritage of the language. The system is started by typing HELLO and ended by typing BYE, files are called up by the OLD command, and unrecognizable command errors produce the response, "WHAT?" The screen presentation, however, is all modern. Initially, the top 18 lines of the screen display the loaded program, and the bottom 7 comprise a history window where immediate commands are typed. The boundary between the windows can be changed at will. The history window can be scrolled to review previous commands, but unfortunately these commands may not be reissued by pointing to them and pressing the Enter key. Only the current command line at the bottom of the history window accepts keyboard input.

The upper editing window is used to enter or modify the program. The editor's key-driven command structure is logically designed, and its response time is acceptably quick. Blocks of lines, with or without line numbers, may be moved or copied, and find and replace commands are implemented.

True BASIC compiles source programs to an intermediate code that executes much more quickly than BASICA's tokenized format. Programs may be saved in either source or compiled form, but the latter still requires the True BASIC system for execution. An optional runtime system is available for converting True BASIC programs into .EXE files that may be run from DOS.

True BASIC implements two features that were a hallmark of the original time-sharing BASIC. The first is a single numeric type for all values. BASIC was designed as an untyped language; integers were added to speed it up on microcomputers. As shown by True BASIC's winning performance in arithmetic benchmarks (see below), the problem of speed seems to have been solved in this implementation.

The second feature is one for which mainframe BASIC was justly famous: matrix operations. Single statements are used to read and write arrays, copy one array to another, add arrays, and multiply them by scalars. Matrix multiplication, inversion, transposition, and calculation of determinants are provided for two-dimensional arrays. Functions are available to generate specialized arrays such as constant, identity, and zero matrices.

True BASIC implements both internal and external subprograms. The internal variety implements global variables only; local declarations are not supported. Conversely, variables in external subprograms are strictly local. Parameters to procedures can be passed either by value or by reference, but function parameters are passed by value only. Because passing by value involves making a copy of the argument, passing large arrays as arguments to functions exacts a penalty in execution time and memory usage.

Any type of subprogram can be invoked recursively, but in practice, recursive internal subprograms are not very useful because they lack private variables. External subprograms can be appended to the programs that call them, or they can be collected into library files. In the latter case, the calling program names the libraries it needs, and the libraries are loaded along with the calling programs; libraries cannot be dropped from memory when they are no longer needed.

Although it does not support BASICA's DRAW statement, True BASIC provides an even better graphics capability through its PICTURE statement.

The graphics capabilities of True BASIC are quite impressive. As in BASICA, a full complement of commands is provided for drawing points, lines, boxes, arcs, and circles. But the location of any graphic element is specified in terms of user-defined coordinates, not in terms of pixels. By default, the coordinate origin is in the lower left corner as on graph paper, not in the upper left as in BASICA, and each dimension runs from 0 to 1. These dimensions may be easily changed, and the direction in which the dimension increases may be reversed. This means that the BASICA pixel-based coordinate system may be easily simulated.

Tailoring the screen coordinates to the data results in very easy graphing. For example, when plotting annual data for the years 1980 through 1986, the horizontal dimension may be defined as 1980-1986, and True BASIC will automatically put equal space between the data points along the horizontal axis.

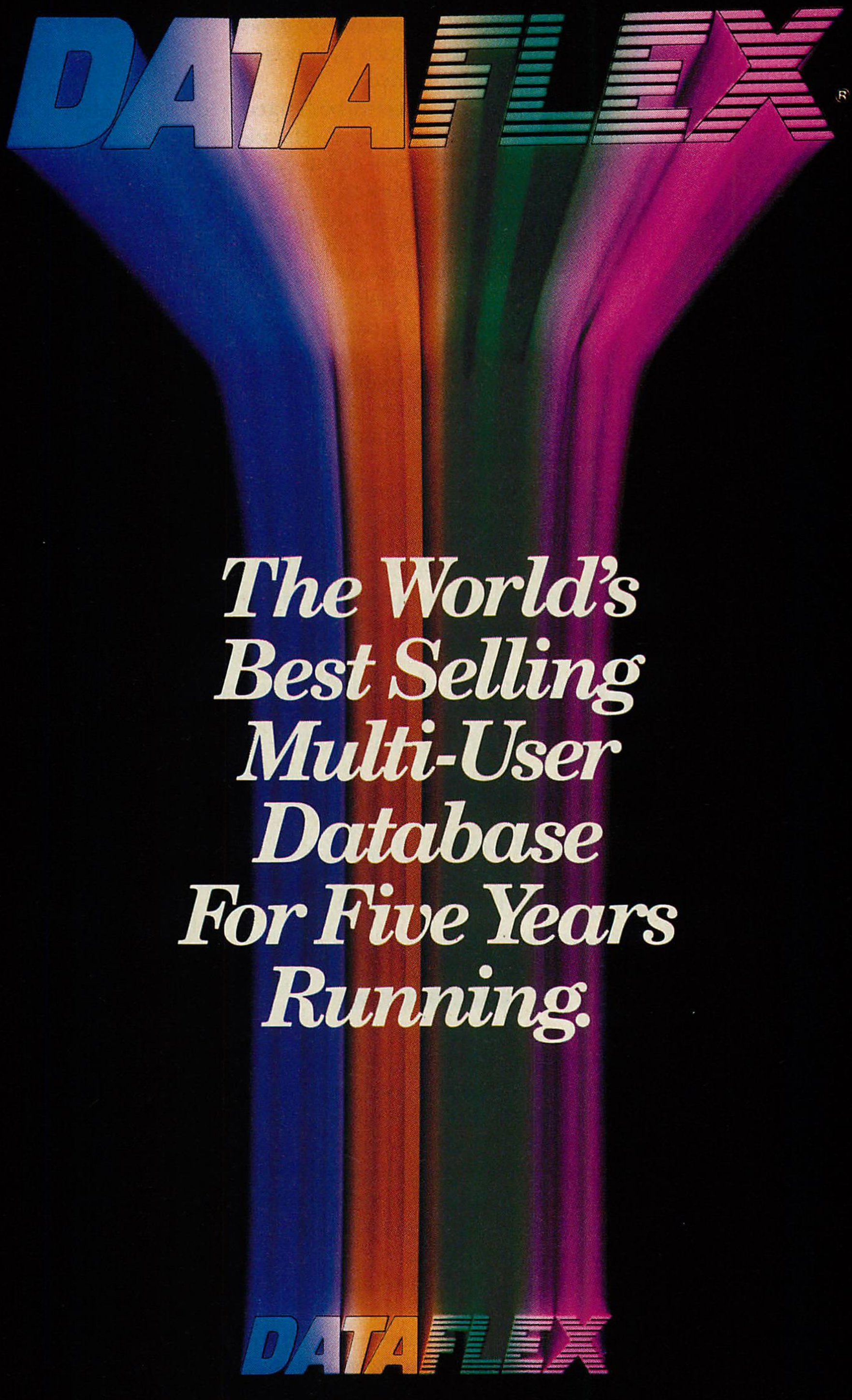
True BASIC permits the definition of windows (both text and graphics) anywhere on the screen. For graphics, each window has its own user-defined coordinates, so the programmer need not be concerned with window size in terms of pixels. When text is written to a window with PRINT statements, its location must be specified in character rows and columns, but text also can be plotted at any position in terms of the window's graphic coordinates. This allows printing of text beginning at any specified pixel location.

Although it does not support BASICA's DRAW statement, True BASIC provides an even better capability through its PICTURE statement. A PICTURE is a subprogram of graphics statements with parameters whose value may change at each call, thereby allowing the user to draw one shape in a variety of sizes, positions, or orientations.

File I/O is True BASIC's weak point. Three kinds of files are supported: text, byte, and record. Text files are implemented as in BASICA, with variable-length records terminated with the CR/LF sequence. For byte files, an arbitrary number of bytes can be read into or written from a string. The implementation of record files is at odds with True BASIC's otherwise modern design. Each fixed-length record can contain only one item (scalar, string, or array element). When writing items shorter than the record size, the remainder of the record is unused, and each ensuing output item goes into a new record.

Record spanning (the breaking up of a long data item over several shorter file records) would be a useful addition with fixed-length records. An error message is generated if an attempt is made to write an item that is longer than the file's declared record size. In most languages that support record-oriented I/O, a record is a collection of several items of different types; in True BASIC, however, such a collection would have to be read and written as several records, each of a length equal to the longest of the items.

In fact, several items can be combined into one string and written into one record, but the method to do so is rather awkward. Like BASICA, True BASIC can convert numerics to and from


DATAFLEX®
The World's
Best Selling
Multi-User
Database
For Five Years
Running.
DATAFLEX

strings. However, it lacks some of BASICA's aids, such as the FIELD statement to assign names to various portions of the record and LSET/RSET to guarantee that fields have the proper length. Therefore, fields must be combined by concatenation and broken apart with substring statements.

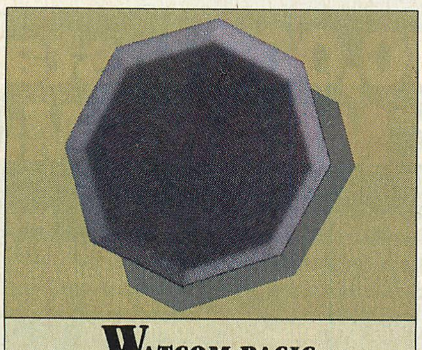
Compatibility with BASICA is not one of True BASIC's aims. An optional program, PC Converter, processes most BASICA statements into True BASIC format, but significant cleaning up with an editor is still required. Statements that have no counterparts in True BASIC are flagged with asterisks to generate compiler errors when the program is first run. Other statements are implemented as subprogram calls, and a subprogram library is provided with the converter program.

Unfortunately, several of the BASICA statements not converted, such as DRAW and FIELD, are those that do the most work and require the greatest effort in order to reconstruct them in True BASIC syntax. Except for DRAW, graphics statements are converted, and the statements to set the screen coordinates to BASICA's screen dimensions are automatically inserted.

Other extra-cost options available for True BASIC are communications support, a developer's toolkit, Btrieve interface, sort/search routines, and Formlib, a screen forms editor.

Documentation consists of two spiral-bound books encased in a flimsy cardboard box that is unable to stand on its own. One of the books is a reference manual, which is organized by subject, not alphabetically. Finding information on specific program statements or functions requires much page flipping. The other book is a user's guide specific to the IBM PC. It presents material of increasing complexity in four sections entitled "Novice", "Intermediate", "Advanced," and "Technical." This last section is a little light on useful information for the technically oriented user. For example, it states that True BASIC uses two formats for floating-point numbers, depending on whether or not the system has an 8087 chip, but it neglects to describe either one of these formats. Otherwise, the documentation is adequate for a user with moderate programming experience. The writing includes many examples.

Except for the poor handling of record file I/O, True BASIC is an excellent implementation of the language and is recommended for BASIC programmers who need more capabilities than are offered by BASICA.



WATCOM BASIC

WATCOM, the company spun off from the University of Waterloo in Canada, is noted for easy-to-use, bulletproof language products especially designed for students. Its BASIC interpreter follows the company's tradition: it is simple, with tutorial documentation, and it is full of structured concepts.

WATCOM BASIC is a true interpreter like BASICA and MEGABASIC. Although it preprocesses programs into a tokenized format, it does not report errors until execution reaches a statement with an error. The program editor is very similar to the one in BASICA; in fact, the only difference noted was the fact that Ctrl-arrow keys do not move the cursor by word. Program files may be loaded and saved in either tokenized or ASCII format, but WATCOM BASIC cannot automatically determine the type of input file. A different command is necessary for loading each type of file: OLD for ASCII, LOAD for tokenized. Similarly, two commands are available for saving each type of file: SAVE for ASCII, STORE for tokenized. Because these commands are difficult to remember, they should be easy to find in the WATCOM manual; unhappily, they are not.

WATCOM has produced the first BASIC that makes a distinction between lower- and uppercase letters in variable names. Fortunately, key words are still case-insensitive.

Numbers may be 16-bit integers or floating-point reals with either single or double precision. Integers are declared by appending % to the variable name; declarations by initial letter (DEFINT) are not possible. This makes conversion of all-integer programs, even simple ones, a much bigger undertaking than it should be. The choice of short or long precision for real numbers is made once at the beginning of the program, so all reals must be of the same precision. WATCOM BASIC automatically detects and uses an 8087 processor.

Matrix operations are supported for initialization (to a scalar or constant matrix such as the identity matrix), assignment, addition of and multiplication by a scalar, addition and multiplication of matrixes, transposition, and inversion. The calculation of determinants is not included. True BASIC is the only other enhanced BASIC product that supports matrix operations.

WATCOM BASIC provides structured constructs in the form of multiline IF-THEN-ELSE statements and several loops. Call-by-name procedures and functions are available; they are internal, meaning that they must reside in the same source file as the main program. All variables within a source file, except for call parameters, are global unless they are declared as local to a subprogram. Old-style GOTO, GOSUB, and single-line IF statements are also supported by WATCOM BASIC.

The file system supports text and binary files. The former are fairly standard, but the implementation of binary files has several annoying quirks. The set of file I/O statements is asymmetrical; a GET statement is used to read single bytes from a binary file, but there is no corresponding PUT. Binary writes are performed with the PRINT statement using a binary format code. For the GET statement to work, either a record length must be specified when the file is open, or the file must contain CR/LF pairs as record delimiters. If a record length is declared, say at ten bytes, then a CR character is returned by

WATCOM BASIC is the first implementation that makes a distinction between lower- and uppercase letters in variable names; key words are still case-insensitive.

every 11th GET, regardless of whether or not a CR byte actually exists in the file. If a CR byte is found in the file, the next character is ignored on the assumption that it is a line feed. Furthermore, all file output operations append CR/LF after each write unless special precautions are taken when the file is opened. This file system makes it very inconvenient to process unstructured binary data to or from programs that are outside of WATCOM BASIC.

Record formats are specified in USING strings. This seems to be a good

To Over 30,000 Installations, MultiLink® MEANS Multi-User.

At over 30,000 sites, as many as one-quarter of a million users tap into the power of MultiLink® Advanced everyday.

Since 1981, they've come to rely on our multi-tasking, multi-user system for compatibility with their favorite software, and the ability to share disks, files, printers, and programs in a *true* PC-DOS environment.

From the largest of the Fortune 500 to the smallest in small business, MultiLink® has provided a cost-effective multi-user solution that's available from no one else.

MultiLink® Means Cost-Effective Timesharing on a PC. MultiLink® Advanced utilizes the principle of timesharing by sharing a central PC's peripherals, files, and processor time among nine users. Up to eight inexpensive terminals can be connected to a single non-dedicated IBM PC, XT, AT or 100% compatible using standard RS-232 ports. Each terminal effectively emulates a PC having up to 512K RAM.

PC-Shadow™ Workstations, shown below, even have an AT look-alike, as well as work-alike, keyboard, display, and serial port. In addition, password-protected remote access via modem can be made with either dumb terminals or PCs running our terminal emulation software.

MultiLink® Means PC-DOS Compatibility with a *Software-Driven* System. Lotus 1-2-3, Symphony, WordStar, dBASE III, & Multimate are just a sampling of the wide variety of PC-DOS software that's fully compatible.

Our software-driven system is also IBM NETBIOS compatible, so programs that are written for IBM's Token Ring will run on our multi-user system, as well.

MultiLink® Means Multi-User to Leading Computer Publications. Whether you read *PC Magazine*, "MultiLink® Advanced delivers on...convenience, speed, and flexibility," or *InfoWorld*, "If you want a low-cost multiuser system with up to eight terminals, MultiLink® Advanced is worth a serious look," it becomes clear that MultiLink® Advanced is a formidable contender in the multi-user marketplace.

See What MultiLink® Can Mean to You. Learn, firsthand, how our multi-user system can benefit your company. Call The Software Link TODAY for complete information and the authorized dealer nearest you. MultiLink® Advanced is $495 and comes with a money-back guarantee.

MultiLink®
ADVANCED

MultiLink® is a registered trademark of The Software Link, Inc. MultiLink® Advanced & PC-Shadow™ are trademarks of The Software Link, Inc. IBM PC, XT, AT, PC-DOS, Token Ring, & NETBIOS are trademarks of IBM Corp. WordStar, dBASE III, Multimate, Lotus 1-2-3 & Symphony are trademarks of MicroPro, Ashton-Tate, Multimate International, & Lotus Development Corp., respectively.

THE SOFTWARE LINK, INC.

8601 Dunwoody Place, Suite 632, Atlanta, GA 30338 Telex 4996147 SWLINK

CALL: 404/998-0700

Dealer Inquiries Invited

THE SOFTWARE LINK, INC./CANADA
250 Cochrane Drive, Suite 12 Markham, Ontario L3R 6B7
CALL: 416/477-5480

CIRCLE NO. 196 ON READER SERVICE CARD

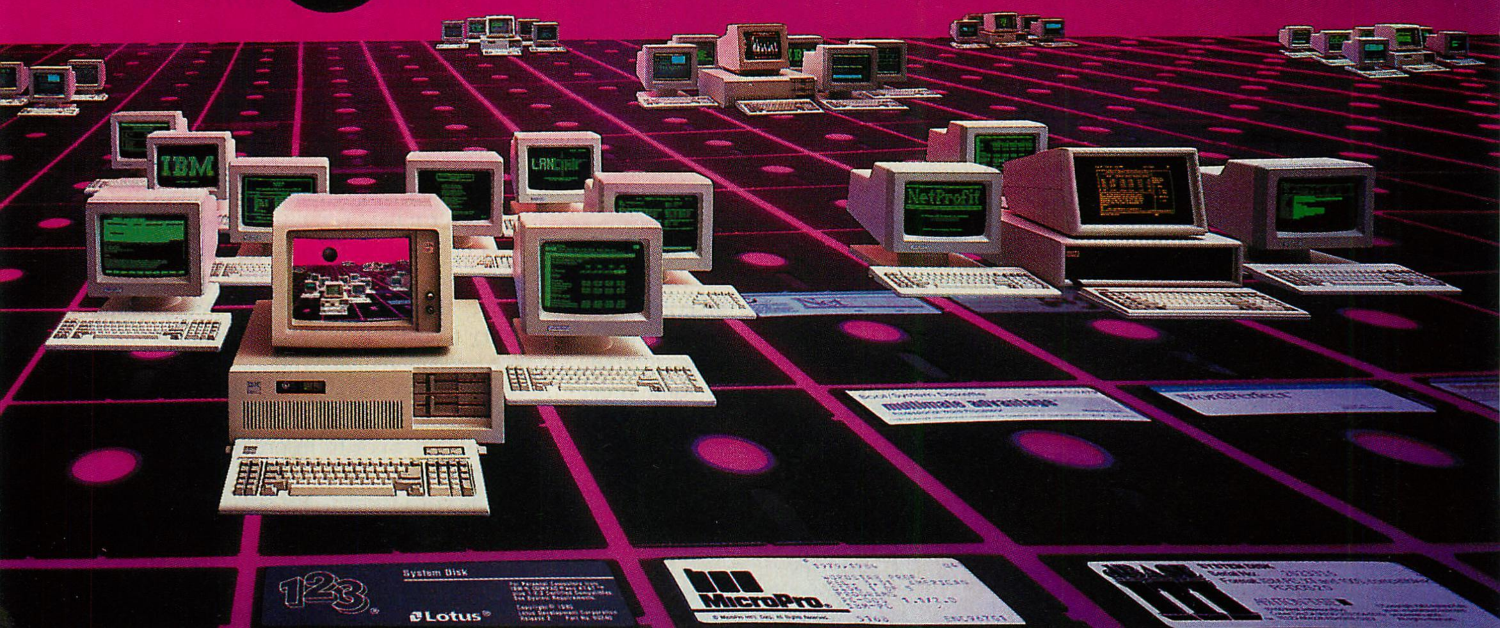


TABLE 2: *Benchmark Times*

	BASICA	BETTER-BASIC	BUSINESS BASIC	MEGA-BASIC	PRO. BASIC	TRUE BASIC	WATCOM BASIC
INTEGER ARITHMETIC							
SIEVE (min:sec)	3:23	0:49	—[a]	1:48	1:25	0:26	4:17
REAL ARITHMETIC							
MULDIV (min:sec)	4:12	1:11	3:29	1:12	4:18	0:20	1:40
Calculation error	0	0	0	0	—[b]	3E-14	3E-16
FILE I/O							
FILEIO (bytes/sec)	112	16	100	408	98	20	71
Minutes to copy 30KB file	9	62	10	2.5	10	50	14
GRAPHICS							
HAT (min)	53	36	—[c]	—[c]	147	61	64

[a]*Program too large for workspace* [b]*Bug prevents print-out of error value; see text* [c]*Graphics support not implemented*

Benchmark surprises are the speed of True BASIC's untyped numbers and BASICA's respectable showing in file I/O.

idea except that "10 AS WHATEVER" in a FIELD statement is much more readable than a string of ten # characters in a format string. Two further problems are that numeric data cannot be written in binary format, but are always converted to/from characters on I/O, and that the fields of the record must be enumerated in each I/O statement. Because record layouts are distributed throughout the program, updating them becomes a chore.

WATCOM BASIC provides most BASICA graphics capabilities, but all of the functions are performed with different key words. Especially frustrating is the fact that a function, not a command, is required to switch between graphics and text modes. The user must type X=MODE(0) instead of BASICA's SCREEN(0). A stripped-down version of the DRAW statement is implemented, without the capability to scale images or execute substrings. Sound and event trapping are not supported. WATCOM BASIC cannot be used to write an interrupt-driven communications program. The interpreter itself, however, supports the serial port as a device for reading and writing program files.

The documentation consists of two spiral-bound booklets. The *Primer and Reference Manual* is primarily a teaching tool and makes a very poor reference. The reference section is organized by subject. Besides the index, the only alphabetic list in the book is an appendix of all immediate commands and program statements, but only their syntax is shown—nothing is said about what each one does. A user deciding whether to save a program with SAVE or STORE may spend several frustrating minutes flipping through pages before finding the answer.

The other booklet is a *User's Guide* specific to the IBM PC, which discusses file I/O, graphics commands, and internal data representations. This part of the documentation is more usable than the reference manual.

WATCOM BASIC is useful as an educational tool, but even with its enhancements in coding structure, it is not a replacement for BASICA's capabilities.

THE BENCHMARKS

Because BASIC is not formally standardized, the number of dialects and the magnitude of the differences between them exceed those in other languages. For this reason, developing a full suite of benchmarks was impossible. Four programs were used to get a general idea of the efficiency of each BASIC for various types of processing. The results are listed in table 2. All timings were done with the built-in BASIC timer and exclude any precompilation time.

The standard Sieve of Eratosthenes algorithm, performed with a matrix size of 8,191, tests integer arithmetic. BB^x could not run this test because it implements all numbers as 10-byte decimal floating point, but it allows a maximum workspace of 64KB.

Speed and accuracy of floating-point arithmetic was tested by the MULDIV program, translated from the Turbo Pascal version published in the April 1986 issue of *PC Tech Journal* (see "8088 vs. V20," Juan Jimenez and Steve King, p. 73). It was performed in the greatest precision offered by each of the BASIC products.

The FILEIO program simulates disk-intensive activities by copying single bytes from one file to another. This is not the most efficient way to copy files, but it is not meant to be a test of file copying efficiency. Single-byte I/O is very common in BASIC, as shown by the many published programs that build .COM files from DATA statements.

Both input and output files were on the same floppy disk. The test was run on a floppy-disk drive for two reasons. First, BASIC is often used on a system without a hard disk. Second, timings on a floppy disk are more consistent than on a hard disk, because hard disk models are more numerous and variable than floppy drives and hard-disk I/O time is affected by location of free space on the disk. On a floppy, the two files can be kept in a constant spatial relationship to each other, and any user can duplicate the conditions.

Several of the BASICs allowed more than one formulation of the I/O statements in this program. All of the obvious ones were tried, and the fastest result is reported.

The last benchmark program, HAT, exercises graphics capabilities. HAT is a good example of programming for portability. The graphics statements are collected into one subroutine, and only the simplest drawing commands (for points and lines) are used. As a result, converting to the syntax of another BASIC is quite trivial. (HAT was originally published in "A Diversionary Benchmark," Susan Glinert-Cole, July-August 1983, p. 95.)

All 19 programs run in testing the BASIC interpreters are available on PCTECHline; three of them are printed with this article. Listing 1 (FILEIO.BAS) is the BASICA version of the I/O benchmark. This program ran as-is in BetterBASIC. However, it ran much faster with two changes (the GET and PUT statements); of its own volition, BetterBASIC added many declarations (see listing 2,

FILEIO.BB). The True BASIC version (listing 3, FILEIO.TRU) had to be written from scratch. In these listings, programs are identified by the following extension: .BAS for BASICA and generic programs; .BB for BetterBASIC; .BBX for Business BASIC; .TRU for True BASIC; .MEG for MEGABASIC; and .WAT for WATCOM BASIC. If a particular interpreter does not have its own copy of a program, it can run the .BAS version.

AN OVERALL WINNER?

The results in table 2 contain at least one surprise: BASICA is not totally overwhelmed by the competition in terms of speed. It makes an especially respectable showing in file I/O, comfortably leading the full-featured BASICs and coming in second behind MEGABASIC. In the other tests, True BASIC is fastest in arithmetic, while BetterBASIC has the fastest graphics. No overall winner or loser could be proclaimed on the basis of execution speed.

Nor could an overall winner be named in terms of general features. The choice of an enhanced BASIC ultimately depends on the uses to which a BASIC interpreter will be put and on the particular shortcomings of BASICA that the user seeks to remedy.

BB^x has many flaws and is not recommended. Its major problems are small memory utilization and lack of modern structured programming constructs. Furthermore, it offers no hardware-specific features such as graphics, sound, or communications support, no PEEK or POKE commands, and only limited editing functions. BB^x cannot import BASICA programs—a serious flaw. It does, however, provide a few advanced features: call-by-name subprograms with local variables, keyed files, and decimal arithmetic. Subprograms can be made either transient or resident at the programmer's option.

MEGABASIC is a well-executed implementation of generic MS-DOS BASIC. It lacks PC-specific functions, such as graphics, sound, and communications, and, like BB^x, it cannot accept BASICA programs. Among MEGABASIC's good

THE ANSI STANDARD FOR BASIC

For several years the American National Standards Institute has been working to produce a standard for BASIC. The first attempt was made in 1978 with the issuance of Minimal BASIC. A new standard is nearing the end of the approval process and should be official this summer.

ANSI BASIC is designed to allow a wide variety of applications to be programmed in the language. Ease of use, avoidance of the unexpected, and portability over a wide variety of hardware and operating systems are prime concerns and take priority over ease of implementation in nearly all cases. The design was chosen to be suitable for both interpreters and compilers. The proposed standard consists of a large core with optional modules for graphics, fixed decimal arithmetic, extensions to file handling, editing, and realtime programming (multitasking).

A rich variety of control structures permit structured programming techniques. Many of the implementations that are based on the proposed standard allow programs to be written without line numbers, although this is not required in the standard. The control structures include looping with the possibility of exit anywhere within the loop; multiline IF-THEN-ELSEIF-ELSE blocks; and a flexible case selection mechanism. Some examples:

```
DO
    INPUT X
IF X > 0 AND X <= 7 AND X=INT(X)
  THEN EXIT DO
    PRINT "Enter an integer
  between 1 and 7"
LOOP
```

and

```
SELECT CASE A$(1:1)
CASE "A" TO "Z", "a" TO "z"
     PRINT A$; " starts with a letter"
CASE "0" TO "9"
     PRINT A$; "starts with a digit"
CASE ELSE
     PRINT A$; "begins with neither"
END SELECT
```

Several modes can be expressed in the OPTION statement. The arithmetic type may be floating decimal, fixed decimal, or native—whatever the implementor wants. Trig functions can use either radians or degrees. The default lower boundary for arrays may be 0 or 1, and the implementor might provide a non-ASCII character set in addition to ASCII.

ANSI BASIC also provides several means for segmenting programs. Functions, subprograms, and pictures (graphics routines) may be either internal or external. External routines have their own name spaces. Internal routines share variable names with the main program or external routine in which they are located. Recursive calls of these routines are permitted. External routines are designed so that separate compilation is possible, though not mandated by the standard. Names of routines, like those for variables, may be up to 31 characters long. Implementations must not compromise the uniqueness of names. Thus:

```
VERY_LONG_ROUTINE_IDENTIFIER_1
```

and

```
VERY_LONG_ROUTINE_IDENTIFIER_2
```

must not refer to the same object.

Arrays may be manipulated element-by-element or as a whole. Numeric arrays may be assigned, set to 0, set to a constant value, added, subtracted, multiplied, transposed, and inverted. String arrays may be assigned, set to nulls, and concatenated. Whole arrays may be used with input and output statements. Arrays also are used in significant ways in graphics.

A large portion of the proposed ANSI standard is devoted to files, which are classified by their overall organization and by the type of record they contain. A file organized sequentially contains records that are accessed one at a time, from first to last. A file with a stream organization contains a sequence of values—as distinguished from a sequence of records. Relative files contain addressable record areas. Keyed files can have their records accessed by the value of a record's (string-valued) key. Record types for files include display (ASCII characters), internal (values with distinct data types), and native (values take meaning according to templates used with input and output statements). A BASIC implementation might use internal records to hold real numbers in a specialized binary format or native records to read data files from a certain COBOL compiler.

The standard takes great pains to describe what happens when errors occur. For example, after an attempt to read a string value into a numeric variable, ANSI BASIC defines the position within a file where the next operation would take place. In addition to defining what happens, the standard provides means of handling exceptional circumstances. Input and output statements may tell what happens if the end-of-file is reached or records

points are all the desirable language extensions: full memory utilization, structured coding, modular programming with separate compilation. MEGABASIC offers more extensions for subprograms and string handling than any other BASIC reviewed here, and its documentation is first-rate. Only the lack of machine-specific functions prevents APC's product from receiving a much stronger endorsement.

WATCOM BASIC is more useful in the classroom than in business. It provides the features that are most desirable in a modern teaching language: structured coding and modular programming. Matrix computations, also useful in a scientific setting, are supported. It has an abbreviated implementation of graphics support. Disadvantages are a small workspace (smaller than BASICA's), lack of separate compilation, and documentation that emphasizes the educational aspect of this product at the expense of providing a usable reference. WATCOM BASIC is recommended only for the educational environment; otherwise, both the program and its documentation will be quickly outgrown.

BetterBASIC promises the best of all possibilities: near-total compatibility with BASICA, plus advanced enhancements that depart from BASIC's simplicity but give the language many of the capabilities of Pascal and C. Delivery of its promises is lacking, however.

The product design is impressive, so it is especially disappointing that a series of individually minor faults make using BetterBASIC a frustrating experience. These faults include slow screen response in the editor, inflexibility in defining data types, incomplete implementation of the DRAW statement, and several quirks during loading of files and navigating around the screen.

The best feature of BetterBASIC is that it addresses all of BASICA's shortcomings: large memory utilization, structured coding constructs, call-by-name subroutines with local variables, and separate compilation. It adds features that go far beyond the capabilities of BASICA: pointers and structures (the

---

are missing. In these cases, programs may exit loops or transfer to specific lines. A control structure for more general exception handling is provided. For example:

```
WHEN EXCEPTION IN
    PRINT "Enter age and weight"
    INPUT AGE, WEIGHT
    IF AGE > 10 THEN
    PRINT "Enter height"
    INPUT HEIGHT
    END IF
USE !COME HERE ON EXCEPTIONS
    PRINT "Enter numbers only"
    RETRY !TRY INPUT AGAIN
END WHEN
! COME HERE WHEN ALL IS WELL
```

The optional graphics module of ANSI BASIC is based on the IBM Graphical Kernal System (GKS) standard. Implementations of GKS for other languages is done with CALLs to a subroutine package. In ANSI BASIC, access to much of GKS is accomplished through statements in the language. Programmers express graphics in problem terms, rather than in screen-dependent coordinates. For example, if a problem called for values in the horizontal direction ranging from 0 to 50 and values in the vertical direction ranging from −10 to 10, then the following statement

```
SET WINDOW 0, 50, -10, 10
```

could be used to express the current range of interest. To draw a line from point (5,5) through (20,6.5) to (23,−4), might require the statement

```
PLOT 5,5; 20,6.5; 23,-4
```

In addition to setting the coordinate system and drawing lines, the user can display individual points, areas, and text with a variety of styles and colors (if supported in the implementation). Graphics is possible in terms of arrays of cells, which may or may not correspond to pixels on a screen. Graphics input from a variety of devices is also supported. Graphics pictures are like subroutines, but the graphics they display may be transformed by matrix expressions on the calling statement. The matrix expressions normally are in terms of the built-in functions ROTATE, SHEAR, SHIFT, and SCALE. For example,

```
DRAW SQUARE WITH SCALE(2,4) *
  ROTATE(45)
```

would call PICTURE SQUARE and change its output so that it would be twice as wide and four times as high; then it would be rotated 45 degrees (if degrees are being used).

The realtime programming module of the ANSI BASIC standard is designed for applications such as process control where it is necessary to have a number of primarily independent activities that are able to communicate with each other.

ANSI BASIC defines no commands other than a small number in the optional editing module. Many of the popular commands were thought to be operating-system-specific and thus beyond the scope of the current draft standard. Microsoft BASIC users will find that the following statements have no equivalent in the standard, but may appear in implementations based on the standard: BEEP, BLOAD, BSAVE, ON COM, ON KEY, ON PLAY, ON PEN, OUT, PLAY, POKE, SCREEN, SOUND, SWAP, WAIT, and WIDTH.

The proposed standard and Microsoft provide approximately the same number of functions, but these Microsoft functions are not in the ANSI version: CDBL, CINT, CSNG, CSRLIN, CVI, CVS, CVD, EOF, FRE, INKEY$, INP, INPUT$, LPOS, MKI$, MKS$, MKD$, PEEK, PLAY, PMAP, SCREEN, SPC, TIMER, USR, and VARPTR($).

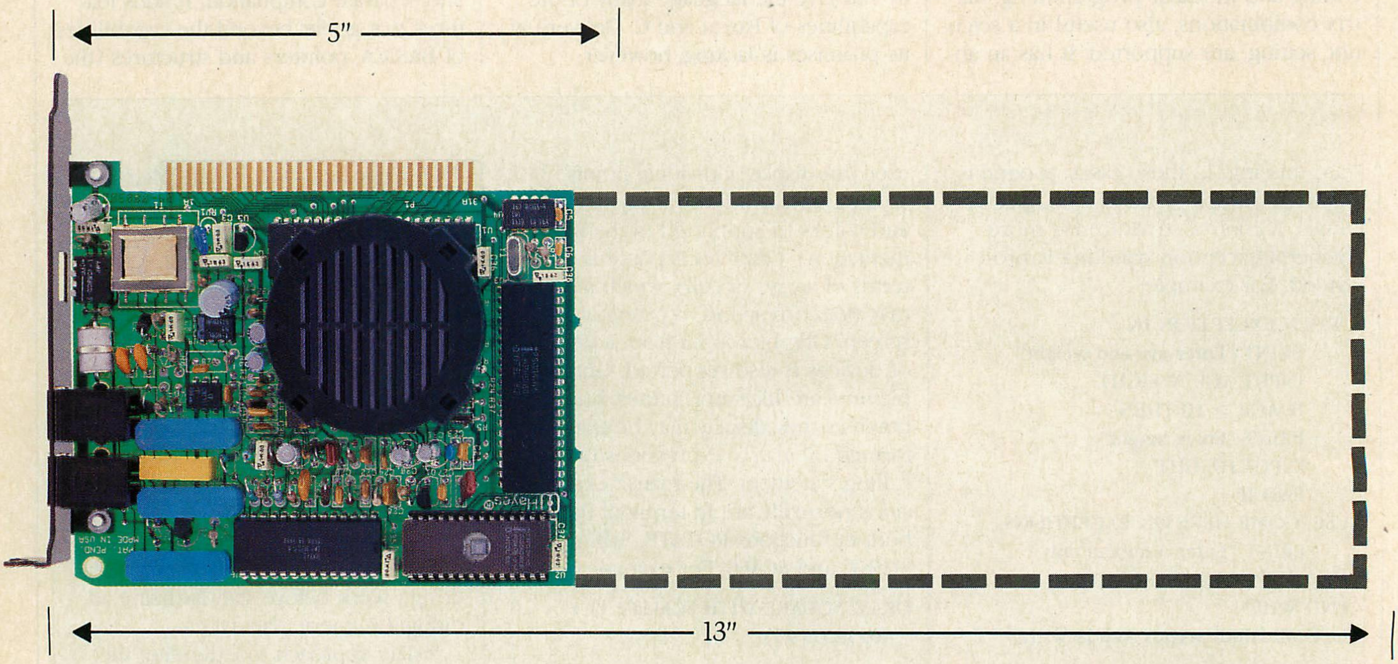
Integers are not included in ANSI BASIC. In fact, the standard does not have a provision for more than one type of number in a given program unit. The ANSI BASIC standards committee wished to do more complete design work before standardizing additional numeric data types.

The approach to extending the standard by adding modules to a central core will allow such extensions to see the light of day before the next complete revision cycle of the language. A mini-standard for single character input for BASIC should be available for public comment this summer or fall. Another standard for modules (similar to that capability in Ada and Modula-2) should follow in three to six months. Work is also being done in these areas: extended array capabilities, data types, data structures, screen management, and sound.

As is the case with nearly all ANSI standards, compliance with ANSI BASIC will be voluntary in the private sector. If the U.S. government adopts a standard as a Federal Information Processing Standard (FIPS), government agencies often require adherence to the standard. ANSI Minimal BASIC is a FIPS, and the proposed BASIC standard is expected to be one as well.

—Jim Harle
ANSI BASIC standards committee

The largest selling 1200 bps modem just got smaller.



The new Hayes Smartmodem 1200B™

Now you can get a lot more out of your PC, by putting a little more in. Our new Hayes Smartmodem 1200B includes the same quality and advanced features that have made it the leading 1200 bps modem. Now, advances in Hayes technology allow us to make it available in a size that fits either full slots or a "single" half slot.

That's important news if you have an IBM,® AT&T,® Compaq,® Tandy® or other compatible computer with half slots. It means with a Smartmodem 1200B, you can free up one of your full slots for an additional function, such as color graphics, more memory or networking. Or, if you prefer, you can continue to use the new Smartmodem 1200B in one of the full slots. Hayes makes it easy and versatile to fit your needs.

There are many good reasons for choosing Hayes. Our new space-saving Smartmodem 1200B is one of them. Hayes Smartcom II,® the industry's best selling communications software, is another.

Smartcom II for the IBM and compatibles makes short work of communicating, while allowing you to take full advantage of the sophisticated capabilities of your Smartmodem 1200B. Together, they create a powerful, yet easy-to-use, communications system for your PC. They're made for each other, and customized for IBM PC's.

The best reasons of all for choosing Hayes are the "built-in" benefits. Advanced technology. Unsurpassed reliability. And a customer service organization that's second to none.

So, when you see your authorized Hayes dealer ask for the largest selling 1200 bps modem. Smartmodem 1200B. And remember. Now it's smaller, too. Hayes Microcomputer Products, Inc. P.O. Box 105203, Atlanta, GA30348. (404) 441-1617.

Hayes®

Say yes to the future

CIRCLE NO. 155 ON READER SERVICE CARD

latter allows the best implementation of record I/O of any BASIC). It may be programmed just like BASICA, or with only a moderate investment in learning, the user may graduate to some very powerful programming capabilities.

Judging by the immense improvement from version 1.0 to 2.0, future updates may well solve some of these problems. Summit Software should keep working to improve this one, because the underlying concept deserves to succeed. BetterBASIC could become an excellent vehicle for any type of programming, including the development of serious commercial applications.

Professional BASIC is supposed to be a development and debugging system for a subset of BASICA, with minor extensions. It is not too useful for complex graphics or any communications program, but it is good for computationally intensive programs (those with complex algorithms and subtle bugs.) A versatile windowing system affords insight into various aspects of a program's execution, allowing rapid zeroing in on runtime bugs. The extensions are useful during the development phase, yet easy to remove when returning programs to execute under BASICA. The language restrictions in comparison to BASICA (no default size for arrays, distinct names for arrays and scalars, and single loop termination statements) are not burdensome; in fact, they foster a desirable programming style.

Of the implementations reviewed here, the one most highly recommended is True BASIC, which addresses most of the limitations of BASICA. It provides large memory utilization, structured coding constructs, local variables, and separate compilation—all in a package that is well thought out and quite easy to use.

Despite these advanced features, True BASIC adheres closely to the informal, unstructured spirit of the original BASIC language. Although True BASIC is somewhat hampered by inconvenient file I/O, it has many strong points, such as 8087 support, fast arithmetic even without the 8087, matrix operations, and an excellent implementation of graphics. This all adds up to make True BASIC the language of choice, especially for graphics and compute-intensive programs.

BASICA: bundled with DOS
IBM Corporation
P.O. Box 1328
Boca Raton, FL 33429-1328
Contact the local IBM dealer; 800/426-2468
CIRCLE 340 ON READER SERVICE CARD

BetterBASIC: $199; runtime: $250
Summit Software Technology
106 Access Road
Norwood, MA 02062
617/769-7966
CIRCLE 341 ON READER SERVICE CARD

Business Basic Extended: $295
BASIS, Inc.
5700 Harper Drive N.E., Suite 290
Albuquerque, NM 87109
505/821-4407
CIRCLE 342 ON READER SERVICE CARD

MEGABASIC: $375
American Planning Corporation
4600 Duke Street, Suite 423
Alexandria, VA 22304
703/751-0451
CIRCLE 343 ON READER SERVICE CARD

Professional BASIC: $99
Morgan Computing Company, Inc.
P.O. Box 112730
Carrollton, TX 75011
214/245-4763
CIRCLE 344 ON READER SERVICE CARD

True BASIC: $149; runtime: $500
True BASIC, Inc.
39 S. Main Street
Hanover, NH 03755
603/643-3882
CIRCLE 345 ON READER SERVICE CARD

WATCOM BASIC: $250
WATCOM Products
415 Phillips Street
Waterloo, Ontario, N2L3X2
519/886-3700
CIRCLE 346 ON READER SERVICE CARD

Ted Mirecki is a corporate planner who is responsible for developing decision support systems on a variety of hardware.

LISTING 1: FILEIO.BAS

```
090 REM          for PC-BASIC & PROFESSIONAL BASIC
095 REM
100 PRINT TIME$
110 KILL "A:OUT.FIL"
120 OPEN "A:IN.FIL" AS #1 LEN = 1
130 OPEN "A:OUT.FIL" AS #2 LEN = 1
140 FIELD 1, 1 AS REC1$
150 FIELD 2, 1 AS REC2$
160 FOR I = 1 TO 30000
170   GET #1
180   LSET REC2$ = REC1$
190   PUT #2
200 NEXT I
205 CLOSE
210 PRINT TIME$
```

LISTING 2: FILEIO.BB

```
SOURCE
PRECISION= 7
AUTODEF=ON
OPTION BASE=0
ERL=ON
ERRORMODE=GLOBAL
RESUME=STATEMENT
FORMODE=GW
SCOPE=OFF
PROCS=0
INTEGER: I
BYTE: X

'MAIN Program:  FILEIO BENCHMARK FOR BETTER BASIC

  100 PRINT TIME$
  110 KILL "A:OUT.FIL"
  120 OPEN "A:IN.FIL" AS #1 LEN = 1
  130 OPEN "A:OUT.FIL" AS #2 LEN = 1
  160 FOR I = 1 TO 30000
  170 READ RECORD #1, I, X
  190 WRITE RECORD #2, I, X
  200 NEXT I
  205 CLOSE
  210 PRINT TIME$

ENDFILE
```

LISTING 3: FILEIO.TRU

```
! FILEIO benchmark for True Basic.
! Written from scratch in True Basic.
!
unsave "A:OUT"
print time$
open #1: name "A:IN", access input,organization byte,recsize 1
open #2: name "A:OUT",access output,create new,organization byte,recsize 1
for i = 1 to 100
  read #1: x$
  write #2: x$
next i
close #1
close #2
print time$
end
```

Diary of an

I traded my 16K machine and all those disks for Lotus® 1-2-3 and 256K. And immediately got the urge to merge. I started by merging regional statements in Maine. But before I could get to Iowa, I ran out of gas.

At 512K, I discovered what the coordinates IV169 looked like. I was so far out there it felt like I was walking on the moon. It didn't take long to find out 512K was nothing more than a walk around the block.

640K! Loads of space until I fell in love with integrated software and was back to cutting up files again. Sure, the other functions make my worksheet more persuasive, but I'm back to slugging in and out disks. Shades of 16K.

Old war stories of how you fought your way up through the memory ranks are great to remember as long as you don't have to relive them. And you don't.

Thanks to one incredible breakthrough. Above™ Board PS.

Above Board is the memory board that drove the Lotus®/Intel/Microsoft® expanded memory specification, Infoworld's 1985 Peripheral Product of the Year.

From Intel. The company that has driven more standards than all of the other chip shops put together.

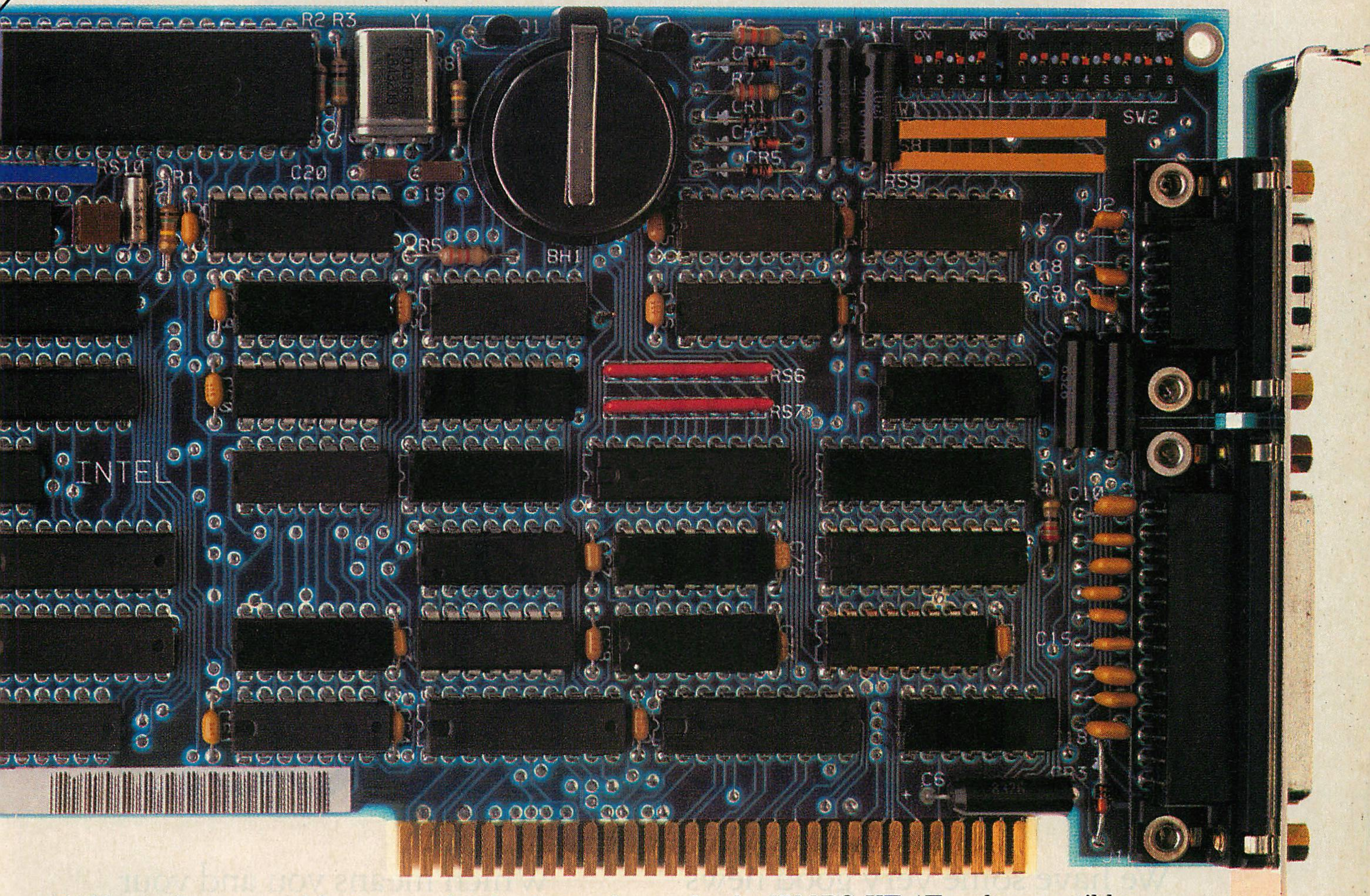
Above Board PS has the parallel and serial ports, clock, print buffer, and RAM disk you'd expect from a conventional multifunction board.

But why buy a conventional board when Above Board PS can take you from 256K all the way to 1.5 megabytes in one fell swoop, without even pausing for a breath at 640K.

So you don't need to hot-wire your way to the top anymore. With Above Board PS, the split worksheet is history. The new norm is windowing, pop-ups, and

overachiever.

I'm on my way to 1.5 MB and I'm not looking back. Monster spread-sheets, killer models, Microsoft® Windows, and all kinds of pop-up utilities so I can juggle a bunch of balls at once.



RAM speed on everything—even databasing and word processing.

The second-biggest difference between Above Board PS and conventional multifunction boards is the unconventional warranty on it. Five years instead of the usual two.

Ease of installation is nothing to sneeze at either. Its software installs with as few as three keystrokes. It even intelligently customizes your memory allocation.

Above Board PS is part of a family of products for the IBM PC, XT, AT and compatibles.

You can buy them at your favorite computer store. Or call 800-538-3373 for the names of dealers near you.

And re-write the book on overachieving.

Help for the overachiever.

intel®

CIRCLE 216 ON READER SERVICE CARD

Ada just moved into a smaller place.

We have some very good news for you.

You can now get a validated, full Ada® compiler for the IBM® PC AT. From the people who designed the Ada language. For just $3,000.

Which means you and your company can now program in Ada without tying up a big, expensive computer.

And you should program in Ada. And not just because the DoD says so.

alsys

1432 Main Street, Waltham, MA 02154

☐ Send me more information about the Alsys PC AT Ada compiler.

☐ Send me more information about Ada.

Name

Title

Company

Address

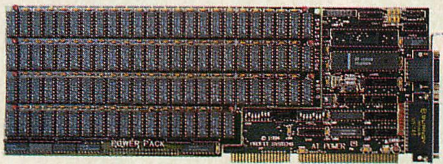
City State Zip

Telephone

The DoD mandates Ada for their software principally because Ada is considerably easier and less expensive to maintain.

Does more reliable and easier to maintain code sound attractive to you? If not, just look at how your programmers are spending 80% of their time.

People who know Ada are calling it "the only logical language for the eighties and nineties." The point is, they're not thinking of Ada as "the DoD language." It's simply because Ada supports good, solid software engineering practice.

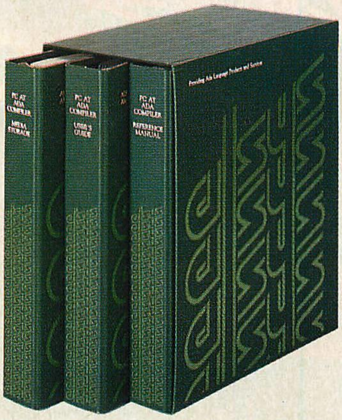


Ada & 4-MB of memory* for the price of 4-MB of memory.

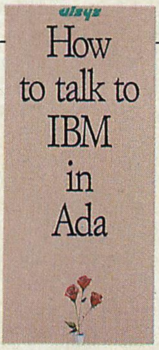
And now you can try full Ada programming for less than the cost of a two week training program.

The Alsys™ Ada compiler for the PC AT is not only validated, it's actually written in Ada. And produces code so efficient it executes faster than C or Pascal on tested benchmarks.

And if that's not enough, the Alsys PC AT Ada compiler runs in protected mode. So you can use the full amount of memory available to the PC AT. This means you can run a program using up to 16 megabytes of memory for code and data without worrying about DOS, overlays and all that stuff.



Lastly, this compiler comes with a 4-megabyte memory upgrade board. That, by itself, is worth the price of admission.



Knowledge is good. Especially when it's free.

At this point, we suspect you might be tempted to pull out a credit card and give us a call.

Or at least fill out the coupon.

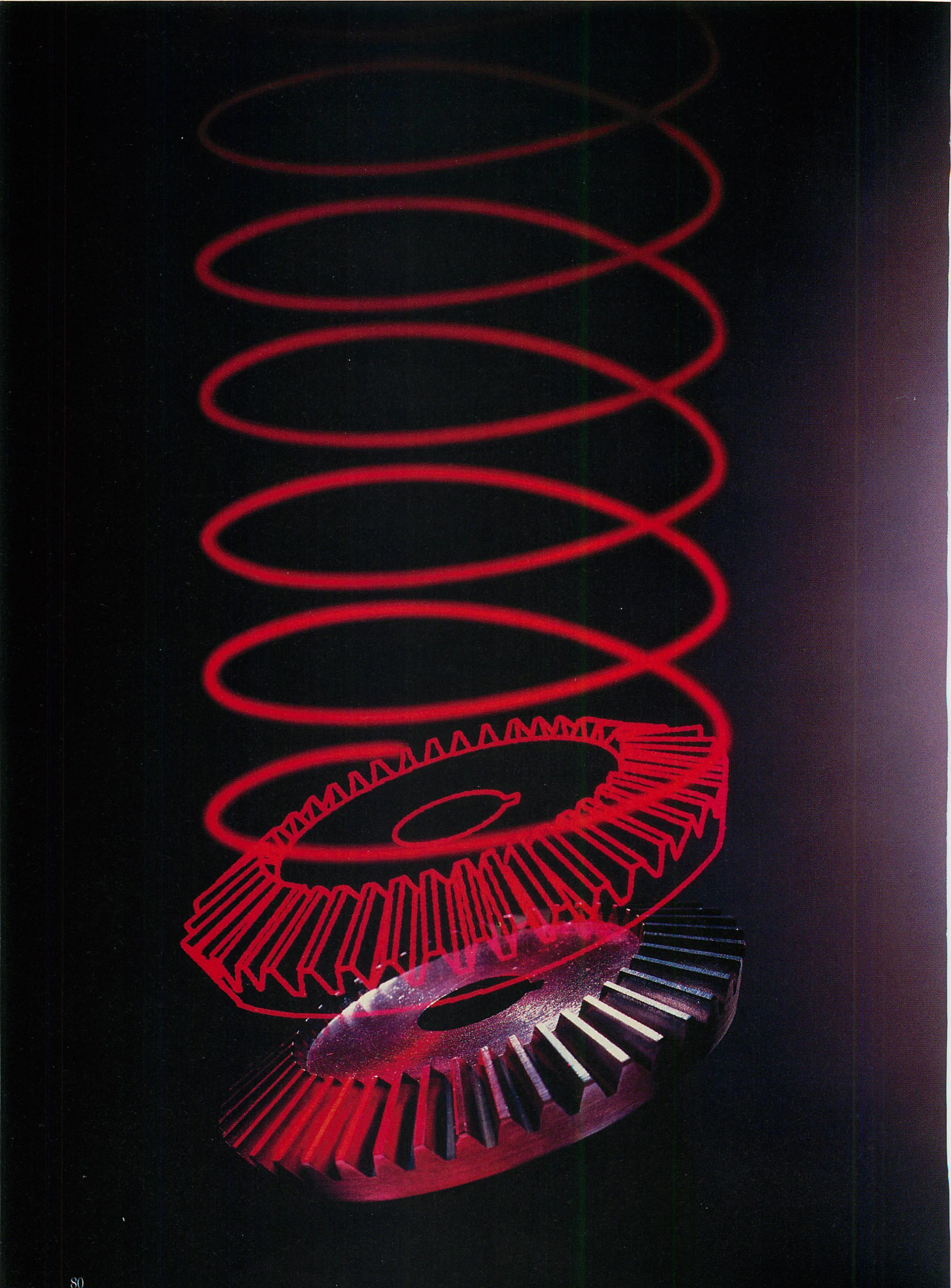
Write: Alsys, Inc., 1432 Main Street, Waltham, MA 02154, U.S.A., Telephone: (617) 890-0030, Telex: 948536.

In France: Alsys, S.A., 29, Avenue de Versailles, 78170 La Celle St. Cloud, France, Telephone: 33(1)3918 12 44, Telex: 697569.

In England: Alsys, Ltd., Partridge Hse, Newtown Road, Henley-on-Thames, Oxon RG9 1EN, England, Telephone: 44 (491) 579090, Telex: 846508.

CIRCLE NO. 200 ON READER SERVICE CARD

Ada® is a registered trademark of the U.S. Government (Ada Joint Program Office). IBM® and PC AT are registered trademarks of International Business Machines Corporation. Alsys™ is a trademark, service mark, and trade name of Alsys S.A. *Memory board manufactured by Profit Systems, Inc.



Mechanical CAD

CADKEY's three-dimensional drawing world is best-suited to mechanical drafting; however, it offers advantages for other fields as well.

VICTOR E. WRIGHT

As do all CAD (computer-aided drafting) systems, CADKEY, Micro Control Systems' entry into the growing microcomputer production drafting CAD market, displays a window into an electronic drawing world. CADKEY's world is three-dimensional, but differs from other such systems in that it is not limited to the modeling of three-dimensional objects (cubes, cylinders, and spheres). The program also allows the entry of one- and two-dimensional objects in arbitrary planes. CADKEY's drawing world is based on floating-point arithmetic and is virtually unlimited at 999,999,999,999 units in each dimension. Resolution is .000001 units, allowing the construction of drawings of stellar-sized objects detailed to optical precision (millionths of an inch).

In its present form, CADKEY is best suited to mechanical drafting; it lacks features that are useful to other disciplines—double line drawing and perspective views used in architectural drafting; varying width lines needed for printed circuit board layout. Micro Control's market strategy has been to aim primarily at mechanical engineers.

CADKEY 2.02 runs on the IBM PC, PC/XT, PC/AT, and close compatibles. The system must include 512KB of RAM and a hard disk with at least 1.7MB of free space—1.5MB to contain the system and 200KB for workspace. A math coprocessor (8087 or 80287) is advised.

CADKEY supports a number of graphics boards, ranging from the IBM Color Graphics Adapter to high-performance boards such as the Conographic Cono-Color 40 and the Number Nine Revolution card. The program supports monitors from Electrohome, Mitsubishi, and others. Some of the graphics cards require that jumpers be set depending upon the monitor used.

A pointing device is not required for minimal operation, but should be a practical requirement. Various digitizers are supported, including the Summagraphics and GTCO tablets.

CADKEY is supplied on five unlabeled 5¼-inch floppy disks. Disk 1 includes a primitive installation batch file that creates the proper directories and copies most of the files into those directories, prompting for the other disks that it requires. The installation program does not give any indication that the correct disks have been inserted, nor does it check that all of the correct files have been installed.

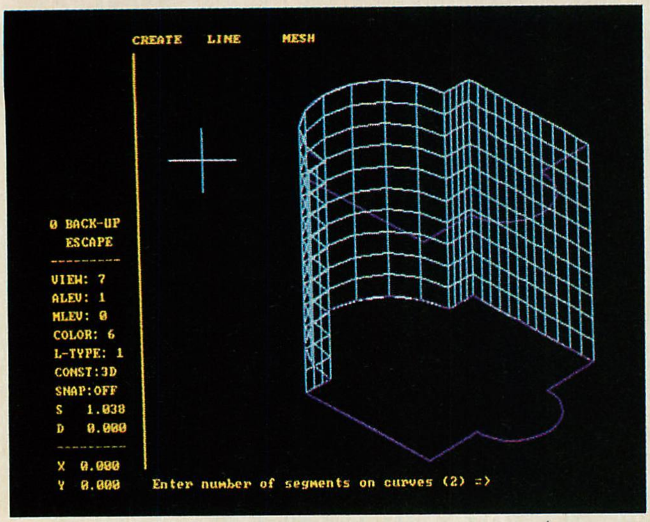
After CADKEY has been installed, it must be configured using the menu-driven CONFIG.EXE file. Selections are made from lists of supported devices with function keys. CONFIG.EXE does not recall the previous configuration, so changing a single element of the system means repeating the entire configuration process. However, special versions or updates of the program are not necessary when the hardware configuration changes in any way.

In addition to setting up the hardware configuration, the CONFIG.EXE program sets default paths to the various subdirectories, which store drawing, symbol, plot, and database files.

CADKEY's disks are not copy protected, and they can be archived, copied, and installed as desired. Included in the CADKEY package, however, is a small box, referred to as the copy protection device, that plugs into the parallel port to ensure that the program is used on only one CPU at a time.

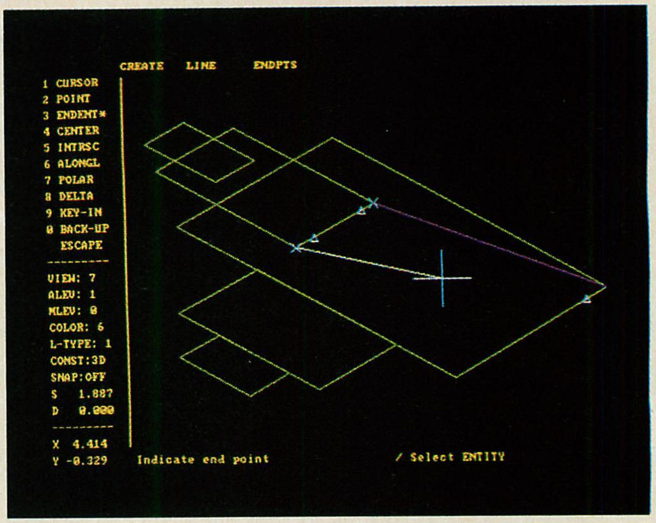
ILLUSTRATION • HERBERT WAGNER COMPUTER IMAGES

PHOTO 1: *CADKEY Screen*



The top line shows the current commands; the current menu options are on the left. The MESH command creates separate lines that can be individually edited.

PHOTO 2: *Position Menu*



CADKEY is able to draw a line from the endpoints of two entities simply by using CREATE-LINE-ENTPTS-ENDENT. A triangle appears on each entity as it is selected.

THE DISPLAY

The CADKEY screen display is divided into two windows: one for drawing display and one for menu selection. Some effort has been expended to leave as much room as possible for the display of drawings, omitting superfluous features such as borders and window frames. The drawing display window occupies the major portion of the screen, while the menu selection window takes up a column along the left side, separated from the drawing display by a vertical bar. In addition to the two major windows, CADKEY displays a prompt/status line at the top and bottom of the screen (see photo 1).

The cursor for the drawing window is a tracking cross, or cross-hair cursor, about one-inch square. It is displayed in white. When the cursor leaves the drawing window to enter the menu area, it changes from a cross to a set of triangles. Menu selections are made by moving the menu cursor to point to the desired item and clicking the pointing device button. CADKEY text color (yellow, by default) and cursor size can be altered with the CONFIG program.

The menu window is divided into three sections. The topmost section displays a current menu or submenu. It contains a maximum of 11 selections, corresponding to the 10 function keys plus the Esc key. These keys can be used to select commands without a pointing device. Two of the keys have a constant value: Esc always returns to the main menu, and F10 (displayed as the 0 key) returns to the previous or next higher level menu.

This top menu area displays a portion of a tree-structured menu. Menu selections near the top of the tree cause submenus to be displayed, listing additional options. As progressively lower menus are selected, the previous commands are also displayed on the top line, in effect displaying the path from the main menu to the current point in the menu structure. In CADKEY terminology, this is the *history line*.

A constant display of options and toggles, provided primarily for control of the display, comprises the second menu area. Selecting one of these puts the current value on the status/prompt line and prompts for a new value, or it simply toggles the option.

The third menu area, located at the bottom of the menu window, displays the *X* and *Y* coordinates of the cursor. The coordinate display functions in two modes: tracking the cursor position continuously or updating the position only when a certain point is selected.

The drawing window can be moved so that the coordinate display overwrites part of the prompt line, and the cursor may be moved to a position behind the top menu line, which is quite irritating. CADKEY could be improved by allowing the pointing device to be active only in areas of the screen where it is functional. A further improvement would be to have the different buttons on the pointing device perform different functions, such as select, back up, or escape, instead of all having the same function.

Entering commands with the menus is fast with a moderate amount of practice; the display changes very quickly, and the menu displays are generated immediately. However, an accomplished user will perceive that a considerable amount of time is required for prompts to be displayed. In addition, using the menus limits the commands that can be issued.

To accommodate the experienced user who knows the command structure and is limited by the menu structure, CADKEY provides an immediate mode in which selected commands can be entered directly, using Ctrl- and Alt-key combinations, regardless of the menu display. Immediate commands are used primarily to change the display during the entry or editing of drawing entities. As an example, the user could enter a long line by zooming in to achieve the desired accuracy, entering the first endpoint, panning with an immediate command, and then entering the other endpoint.

COORDINATES AND VIEWS

The CADKEY manual introduces two types of coordinate systems. *World coordinates* are in the drawing database, relative to the origin of the drawing world. This system is right-handed, Cartesian, and three-dimensional. *View coordinates* are derived from projecting the world coordinate system on the CAD system screen. CADKEY provides eight predefined views, each represented in a right-handed, Cartesian coordinate system: top, front, back, bottom, right, left, isometric, and axonometric. The program begins with the top view displayed, meaning that the *X* and

Y axes are visible—the *X-Y* plane is displayed—and the *Z* axis is perpendicular to the display.

Views 1 through 6 each display two axes in the plane of the screen. The third axis is perpendicular to the screen. View 7 is an isometric view in which distances along all three axes are displayed as being equal. View 8 is an axonometric view in which distances along all three axes are visible, but are not foreshortened equally.

Views are selected for display with the VIEW command, which can be invoked from the top of the menu window, from the status display, or with an immediate command. Views can be called only by number and use the standard numbering system.

In addition to the eight predefined views, the user can define an unlimited number of arbitrary views using VIEW DEFINE. A new view can be specified by naming three points in the viewing plane or by rotating the current display. The program remembers the viewing direction, or viewing plane orientation, and the scale, or zoom ratio, of the display associated with each view.

Views result from rotating and translating a viewing plane relative to the world coordinate system. Drawing entities retain their relationship to the world coordinate system regardless of the definitions of views. CADKEY does not provide a perspective display, but as explained below, it does provide the tools for creating perspective drawings.

The VIEW command is only one of CADKEY's display control commands. The standard commands, ZOOM and PAN, are supplied. Also provided are HALF and DOUBLE commands to halve or double the scale of the display, an autoscaling option to fill the screen with the drawing extents, the BACK-1 command which is used to retrieve up to three previous displays, and the ZOOM WINDOW option.

For those who cannot shake the notion of working at scale, the ZOOM command also has a SCALE option. The program initially presents a drawing display at a scale of 1.000. The SCALE option displays the same prompts as selecting the S status line does. Both prompt for the new scale and view center, offering current values as the defaults. The scale of 1.000 only roughly corresponds to a full-size display. The actual size of a full-scale display depends upon the size of the monitor—a 19-inch model with the same resolution in pixels as a 13-inch model would display the full-size drawing enlarged by a factor of approximately 1.5, with the horizontal and vertical measurements adjusted to the particular monitor.

The entire drawing world cannot be displayed on the screen at one time. The largest scale of the display is limited to .001, even though scale values are displayed on the prompt line to six decimal places.

The PAN command translates the viewing window in the viewing plane without changing the scale of the display. The command is invoked from the DISPLAY menu or from immediate mode. PAN prompts for the new display center, rather than for a displacement vector. The new display center can be selected with the pointing device, or the coordinates can be typed in from the keyboard by the user.

CADKEY includes two commands in the DISPLAY menu that some systems classify as drawing aids: GRID and SNAP. The GRID command displays a grid of dots. The grid spacing can be set either to the snap interval or to arbitrary *X*- and *Y*-axis values. Toggling the grid off does not erase it; instead, the REDRAW command must be issued after the grid is toggled off.

> *In addition to the eight predefined views, the user can define an unlimited number of arbitrary views using VIEW DEFINE.*

The SNAP command has the same options as GRID, with one exception. Whereas GRID provides for setting the grid size to the snap resolution, SNAP provides for setting the snap resolution to the GRID spacing. Thus, the grid can be aligned to a particular part feature, and then the snap resolution can be set to provide snap intervals using that feature as the origin. This is very useful when a part has a number of entities related to a feature, such as the center of a hole, but not necessarily to other features, such as the edges of the part, which may be drawn relative to the drawing's origin.

CADKEY provides a LEVEL (layer) facility for visibility control. The levels are numbered 1 through 256, with level 256 reserved for the storage of objects that are not selected for display.

The display status window includes two status indicators provided for level management: ALEV displays the current active level (the level number assigned to entities as they are created); MLEV displays the masking level (the level from which entities can be selected for deletion). If the masking level is set to 0 (the new drawing default), entities can be selected from any visible level; if the masking level is set to a positive integer, entities can be selected only from that level; and if it is set to a negative integer, entities can be selected from all visible levels except that one.

The LEVEL command has a MOVE option, which allows objects to be moved to a selected level. Entities can be selected on the screen or specified by their current level number.

Note that colors and line types are not related to levels. A level can contain entities of various colors and line types.

CREATING OBJECTS

The basic drawing process begins with the CREATE command of the main menu. CREATE displays eight options: LINE, ARC, CIRCLE, POINT, RECTANG, FILLET, CHAMFER, and POLYGON. Each of these commands has one or more levels of submenus or prompts, and none is available as an immediate command. They can be selected with the menu cursor, via the pointing device, or with the function keys. Thus, in the creation of drawing entities, CADKEY is an entirely menu-driven system.

Missing from the CREATE menu is the polyline entity found in many large CADD (computer-aided design and drafting) systems and at least one microcomputer CAD system, AutoCAD. (For a review of AutoCAD see "Drafting by Design," Victor E. Wright, January 1986, p. 50.) Polylines are lines with attributes other than endpoint coordinates. Some microcomputer CAD systems provide the ability to assign color, width, and line-type attributes to all drawing entities, eliminating the need for polylines. While CADKEY provides the color and line-type attributes to be assigned to all entities, it does not provide line-width attributes. Line width can be specified in pixels for display purposes, but all lines plot as single strokes. While this is not a loss for mechanical engineers, it is for those users working in other disciplines, such as architectural drafting.

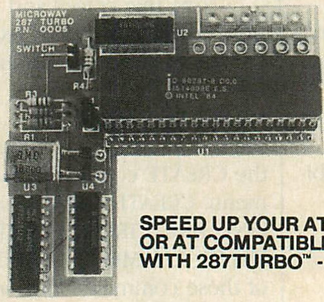
The CREATE <entity> commands operate as modes rather than as commands that must be repeated for the creation of each entity. After descending the menu structure path consisting of the commands CREATE, LINE, ENDPTS, CURSOR, the user can enter an arbitrary

MICROWAY'S 8087 RUNS 1-2-3™!

MicroWay is the world's leading retailer of 8087s and high performance PC upgrades. We stock a complete selection of 8087s that run at 5 and 8 mhz. All of our coprocessors are shipped with a diagnostic disk and the best warranty in the business – one year! We also offer daughterboards for socketless computers (NEC PC) and 287Turbo which increases the clock speed of the 80287 from 4 to 8 mhz. Our NUMBER SMASHER™ includes 640K ram. It will run the IBM PC at clock speeds up to 10 mhz and achieves a throughput of .1 megaflops with 87BASIC/INLINE, Intel Fortran, or Microsoft Fortran. Software reviewers consistently cite MicroWay software and 8087 expertise as the best in the industry! Our customers frequently write to thank us for recommending the correct software and hardware to meet their specific needs. They also thank us for our same day shipping! In addition to our own products which support the 8087 and 80287, we stock the largest supply of specialized software available. For information call us at

617-746-7341

FASTBREAK™ employs the 8087 to increase the speed of Lotus 1-2-3™ Version 1A or 1A* by up to 36 to 1 **$79**
FASTPAK™ includes FASTBREAK software and a 5mhz 8087 .... **$179**
87SFL™ MicroWay's Scientific Function Library contains 170 scientific and engineering functions. Callable from most 8087 compatible compilers. First Language **$250**; Additional **$100**
A2D-160™ MicroWay's Data Acquisition Board performs 160,000 12 bit Analog to Digital conversions per second! Includes software drivers. The fastest 12 bit A to D board available. For the IBM PC or XT......... **$1295**



SPEED UP YOUR AT OR AT COMPATIBLE WITH 287TURBO™ - 8mhz

Micro Way® 8087 Support

For the IBM PC, PC XT, PC AT and Compatibles.

NUMBER SMASHER™ The World's Fastest Accelerator Card for the IBM PC, XT, and Compatibles! Includes an 8086 and 8087 pair tested to 10mhz, and 640K bytes of high speed ram. Compatible with all software, operating systems and hardware! Your program speed is increased by a factor of 2.5 to 4.0. Floating point programs run up to 2.8 times faster on the PC than on an 80287 equipped PC AT. One of the few accelerators on the market which is compatible with all EMS cards. Includes FASTBREAK....... **$1099**

87/88Turbo™ is a stubby card which includes a clock calendar and a speed controller which changes the speed of your motherboard from 4.77 to 7.4mhz. Its use requires your PC to have a socketed 8224. Typical speed increase is 1.6. Some programs run up to 2.5 times faster. The card overcomes the problem of slow hardware by slowing up only when slow devices are accessed and running at full speed otherwise. It includes an 8087-2 .... **$350**

286TurboCache™ This new MicroWay accelerator uses 8K of cache memory and 80286/80287 processors to provide an average speed increase of 3 to 1 for most programs. Call for specifications, benchmarks and price.

MATRIXPAK™ manages a **MEGABYTE!** Written in assembly language, our runtime package accurately manipulates large matrices at very fast speeds. Includes matrix inversion and the solution of simultaneous linear equations. Callable from RM or MS Fortran, MS Assembler, or 87BASIC/INLINE.................... each **$99**

87FFT™ Written in assembly language, performs Forward and Inverse FFTs on real and complex arrays which occupy up to 512K bytes of RAM. Also does convolutions, auto correlations, hamming, complex vector multiplication, and complex to radial conversions. Callable from most 8087 compatible compilers............... **$200**

87FFT-2™ performs two-dimensional FFTs. Ideal for image processing. Requires 87FFT **$100**

Micro Way
P.O. Box 79
Kingston, Mass.
02364 USA
(617) 746-7341

MegaPage™ Our Intel-Lotus specification memory card. The only EMS board which comes with two megabytes of cool-running, low power drain CMOS ram installed. Includes ram disk, print spooler, disk cache, and EMS drivers. For the IBM PC, XT, AT and compatibles............. **$549**

DFixer™ Our disk utility which thoroughly checks PC or AT hard disks for bad sectors and updates the MS DOS file allocation table accordingly. Solves the AT hard disk problem! ... **$149**

DOptimizer™ Optimizes the way your hard disk or floppy stores its files. Speeds up accesses by recombining fragmented files........... **$49**

DCache™ Our disk caching software speeds up your I/O by storing repetitively used tracks in memory. The amount of memory used can be selected in 64K byte banks **$49**

87Verify™ For users who have to be absolutely sure of their results! This background task periodically performs an 8087 accuracy and stress test **$49**

87MACRO/DEBUG™ Contains all the pieces needed for writing 8087/80287 assembly code including 8087 macros, object libraries for commonly used functions, including transcendentals, trigonometrics, conversions and MicroWay's debugger – 87DEBUG **$199**

OBJ→ASM™ A multipass object module translator and disassembler. Produces assembly language listings which include public symbols, external symbols and labels commented with cross references. Ideal for patching object modules for which source is not available............. **$200**

87BASIC™ includes patches to the IBM Basic or MS Quick Basic Compiler for USER TRANSPARENT 8087 support. Provides super fast performance for all numeric operations including trigonometrics, transcendentals, addition, subtraction, multiplication, and division...... each **$150**

87BASIC/INLINE™ converts the output of the IBM Basic Compiler into optimized 8087 inline code which executes up to seven times faster than 87BASIC. Supports separately compiled inline subroutines which are located in their own segments and can contain up to 64K bytes of code. This allows programs greater than 128K! Requires the IBM Basic Compiler Version 1 and Macro Assembler. Includes 87BASIC **$200**

MICROWAY UDI runs RTOS or RMX compilers under DOS **$300**

8087 UPGRADES

All MicroWay 8087s include a one year warranty, complete MicroWay Test Program and accurate Installation Instructions.

8087 5mhz................ $109
For the IBM PC, XT and compatibles.

8087-2 8mhz............ $149
For Wang, AT&T, DeskPro, NEC, Leading Edge.

80287-3 5mhz........... $179
For the IBM PC, AT and 286 compatibles.

80287-8 8mhz.......... $295
For the Tandy 3000 and ITT-XP.

287Turbo™ 8mhz If you own an AT, Deskpro 286 or AT compatible, this is the card you need to get reasonable numeric performance. It plugs into your 80287 socket and includes a specially driven 8mhz 80287. The card comes in three configurations. The IBM AT version includes a hardware RESET button **$395**

NEC V20, V30	**CALL**
64K RAM Set 150ns	**$10**
256K RAM Set 150ns	**$34**
128K RAM Set PC AT	**$49**
JRAM, AST, INTEL	**CALL**
DCache plus 64K ABOVE BOARD PS	375
DCache plus 128K AT ABOVE BOARD	475
Microsoft Fortran Version 3.31	209
IBM Professional Fortran	565
Ryan-McFarland Fortran Version 2.0	399
FORLIB+ or STRINGS and THINGS	65
Grafmatic for Fortran or Pascal	125
Plotmatic	125
NAG Fortran Library	300
Lattice C	269
Microsoft C or QuickBasic	CALL
IBM Basic Compiler Version 2.0	465
Summit BetterBASIC™	CALL
IBM Assembler w/Librarian Version 2.0	155
Microsoft Assembler Version 3.01	99
Microsoft Pascal Version 3.31	199
STSC APL★PLUS/PC	450
Phoenix Plink86 or Cosmos Revelation	CALL
SPSS/PC+	595
EPSILON Text Editor	195
LABTECH NOTEBOOK	745
MultiHalo (one language)	189
Fifth Generation FASTBACK	150

number of line segments without returning to the menu. Similarly, entering the commands CREATE, CIRCLE, and CIR+RAD allows quick entry of a number of circles of the same radius (the program prompts for the radius only once after CIR+RAD is selected).

The CREATE command options lack an orthogonal mode to force the entry of horizontal and vertical lines of specified length. Instead, CADKEY provides separate options for the entry of a vertical or a horizontal line, or one of each, through a specified point. These lines extend from top to bottom and side to side of the current display. A submenu command also is provided to enter lines perpendicular to a reference line, beginning at a specified point.

Several of the commands in the CREATE menu (LINE, ARC, CIRCLE, POINT, and RECTANG) lead eventually to the position menu, which includes these 11 selections: CURSOR, POINT, ENDENT, CENTER, INTRSC, ALONGL, POLAR, DELTA, KEY-IN, BACK-UP, and ESCAPE. These features can be used for snapping to points on entities or groups of entities. For example, a line can be drawn from the end of one line to the end of another by entering the command string CREATE, LINE, ENDPTS, ENDENT, then selecting the two existing lines near the desired ends. The program locates the end of each selected line and constructs a line between those two points (see photo 2).

If the CURSOR option is selected from the position menu, points are entered at the current cursor position and in the current plane—at the current depth. When working in view 1, the top or plan view, keeping track of where objects are being created is a simple matter. The D symbol in the status window displays the elevation or Z-axis distance from the drawing world origin, and the X and Y indicators display the X and Y coordinates of the screen cursor in world coordinates. These coordinates happen to coincide with the view coordinate system. The other views are not so simple. In the front, back, right, and left views, the X and Y indicators do not change to X and Z or Y and Z to display world coordinates. Instead, they continue to display the view coordinates, and the user must keep track of the rotation between the view coordinate system and the world coordinate system.

In isometric, axonometric, and user-defined views, the viewing plane is not perpendicular to any of the three world axes, and the D indicator displays the length of a vector from the world origin to the viewing plane, perpendicular to the plane. A useful addition to CADKEY would be the ability to toggle between the cursor tracking for world coordinates and for view coordinates.

Although CADKEY is a three-dimensional drafting system, it is not a modeling system for solids. CADKEY's drawing primitives are linear objects that can be grouped to form plane shapes. A good example is the MESH option used to model surfaces.

CADKEY can create a mesh of lines or points between two entities. If the selected entities are parallel lines, the mesh models a plane. If the selected entities are lines inclined to each other, the resulting surface is warped. Circles of the same diameter in parallel planes produce surfaces curved in one direction; circles of different sizes produce truncated cones. These surfaces, however, are not stored in the drawing database as identifiable objects. Instead, the individual lines are created and stored. Thus, a surface created with the MESH command is only a series of line segments (see photo 1).

Although CADKEY provides the color and line-type attributes to be assigned to all entities, it does not provide any line-width attributes.

CADKEY's lack of hidden line removal processing and automatic perspective generation also restricts its use as a 3-D modeling system.

EDITING

Because CADKEY allows the active viewing/creation plane to be oriented arbitrarily and provides the ability to snap to features in front of and behind the viewing plane, virtually any shape can be created correctly on the first attempt. Nonetheless, editing is an essential function for any CAD system. CADKEY provides the basic editing functions found in virtually all CAD systems—and a few not found in many of them.

Among the basic editing commands is DELETE, a selection on the main menu. DELETE displays two options: SELECT and LEVEL. SELECT offers four ways to delete objects: SINGLE allows the selection of an arbitrary number of isolated entities; CHAIN provides for the selection of a number of line and arc entities connected end to end; WINDOW deletes objects contained completely within the window, but not those extending outside—the window does not clip objects for partial deletion; and ALL DISP provides a means of deleting all items currently displayed.

DELETE's LEVEL option allows all items on a level to be deleted whether they are currently displayed or not.

An important command related to the DELETE command is the RECALL option of the EDIT command; it is able to reclaim deleted entities (assuming that the deletion has been made in the current drawing session).

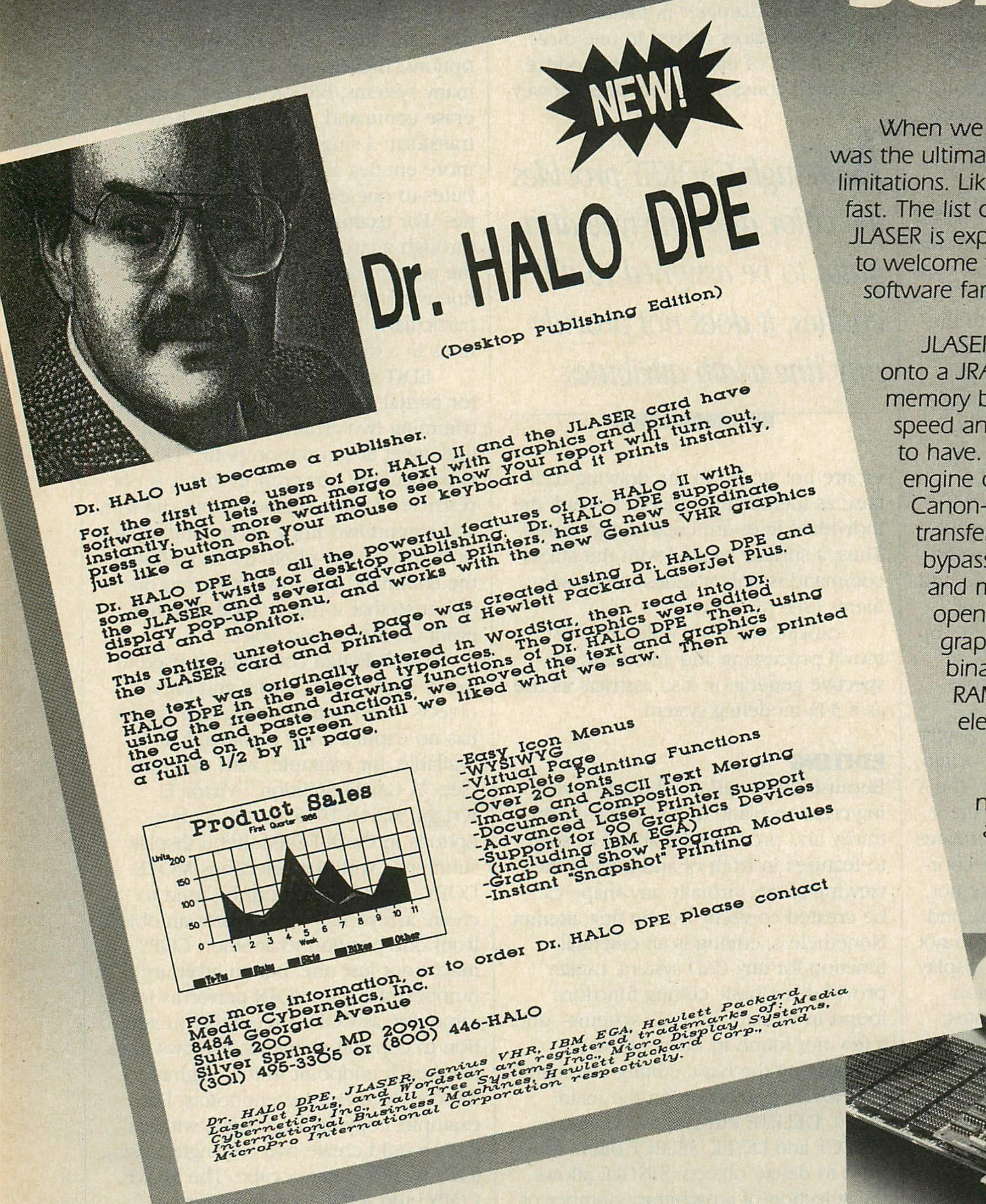
Editing functions are also provided in the EDIT and X-FORM commands of the main menu. EDIT provides two options: BREAK and TRM/EXT. Unlike in many systems, BREAK is not a partial erase command; instead, it is used to transform a single entity into two or more entities and to assign new attributes to one or more of the new entities. For example, a line passing through a circle can be broken so that the portion within the circle changes line type and/or color. This facility is particularly useful for indicating hidden lines in a drawing.

EDIT's TRM/EXT option provides for partial erasure and for extending or trimming two arbitrary lines so that they meet at a common point. TRM/EXT is particularly powerful in that it is not restricted to line segments, but can trim and extend two lines, an arc and a line, or two arcs (see photo 3). In addition, the command can trim or extend one object so that it meets another at any point on the second object.

The X-FORM command is used to move, copy, scale, rotate, and mirror objects (see photo 4). Although CADKEY has no explicit array command, as is available, for example, with CADVANCE (see "A CADD Solution," Victor E. Wright, March 1986, p. 86), its five options in the X-FORM menu display submenus with the selections MOVE, COPY, and JOIN and can be used to create arrays. MOVE translates an object from one position to another; COPY makes not just one, but an arbitrary number of copies. JOIN performs in the same way as COPY except that in addition to copying elements, it draws lines from each endpoint in the original to the copy's respective endpoints. For example, copying a rectangle without JOIN would create two rectangles; using JOIN would create a cube. The MOVE, COPY, and JOIN options, in turn, display menus for the selection of entities by pointing, selecting a chain of

JLASER

THE LASER PRINTER SOLUTION


NEW!

Dr. HALO DPE

(Desktop Publishing Edition)

Dr. HALO just became a publisher.

For the first time, users of Dr. HALO II and the JLASER card have software that lets them merge text with graphics and print instantly. No more waiting to see how your report will turn out, press a button on your mouse or keyboard and it prints instantly, just like a snapshot.

Dr. HALO DPE has all the powerful features of Dr. HALO II with some new twists for desktop publishing. Dr. HALO DPE supports the JLASER and several advanced printers, has a new coordinate display pop-up menu, and works with the new Genius VHR graphics board and monitor.

This entire, unretouched, page was created using Dr. HALO DPE and the JLASER card and printed on a Hewlett Packard LaserJet Plus.

The text was originally entered in WordStar, then read into Dr. HALO DPE in the selected typefaces. The graphics were edited using the freehand drawing functions of Dr. HALO DPE. Then, using the cut and paste functions, we moved the text and graphics around on the screen until we liked what we saw. Then we printed a full 8 1/2 by 11" page.

-Easy Icon Menus
-WYSIWYG
-Virtual Page
-Complete Painting Functions
-Over 20 fonts
-Image and ASCII Text Merging
-Document Compostion
-Advanced Laser Printer Support
-Support for 90 Graphics Devices
(including IBM EGA)
-Grab and Show Program Modules
-Instant "Snapshot" Printing

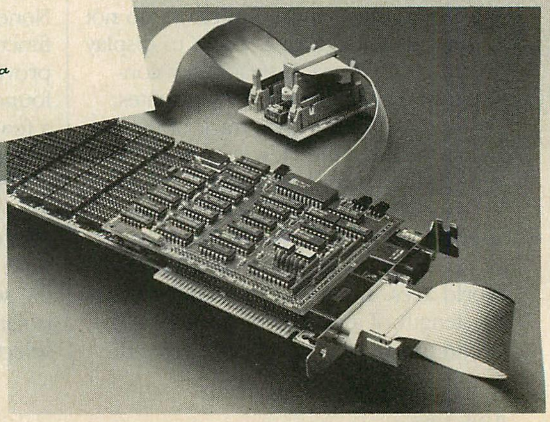
For more information, or to order Dr. HALO DPE please contact
Media Cybernetics, Inc.
8484 Georgia Avenue
Suite 200
Silver Spring, MD 20910
(301) 495-3305 or (800) 446-HALO

Dr. HALO DPE, JLASER, Genius VHR, IBM EGA, Hewlett Packard LaserJet Plus, and Wordstar are registered trademarks of: Media Cybernetics, Inc., Tall Tree Systems Inc., Micro Display Systems, International Business Machines, Hewlett Packard Corp., and MicroPro International Corporation respectively.

When we introduced JLASER, we knew it was the ultimate solution to current laser printer limitations. Like all good things, it's catching on fast. The list of software companies that back JLASER is expanding rapidly. We are pleased to welcome this latest addition to our growing software family of JLASER supporters.

JLASER is a controller that piggybacks onto a JRAM-3 or AT-3 two megabyte memory board giving laser printers the speed and versatility they were meant to have. It plugs directly into the printer engine of the H-P Laserjet™ or other Canon-based printers. Our interface transfers bit-mapped images from RAM, bypassing the on-board print buffer and most of the control circuitry. It opens up a creative new world of graphics and multiple type font combinations, and still gives you plenty of RAM to use for programs, data, and electronic disk.

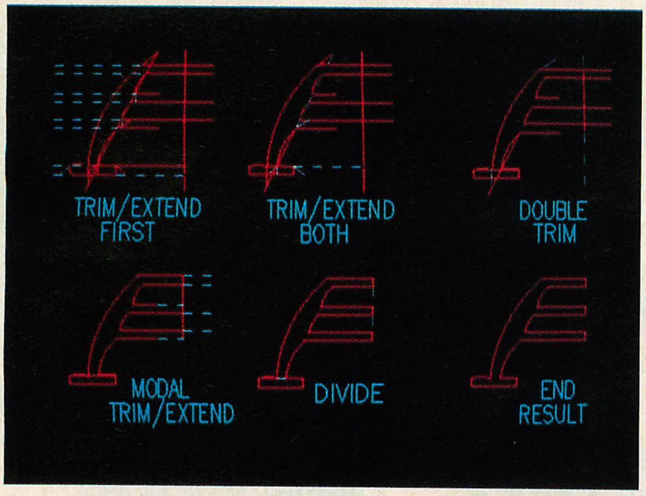
JLASER is the powerful, economical, and efficient solution to all your laser printer problems.



TALL TREE SYSTEMS
1120 San Antonio Road
Palo Alto, CA 94303
(415) 964-1980

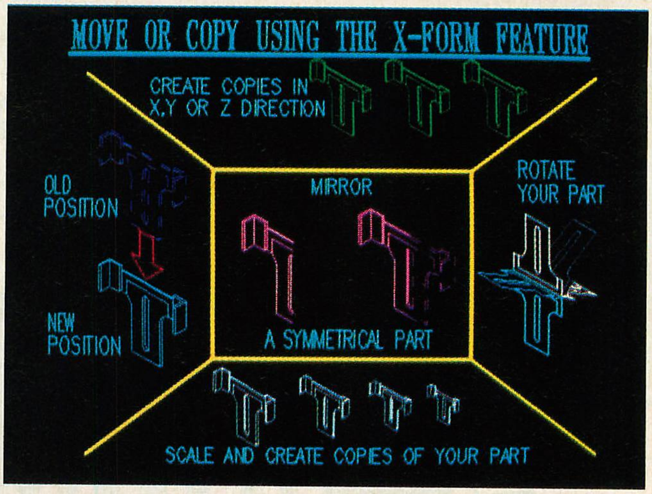
CIRCLE NO. 194 ON READER SERVICE CARD

PHOTO 3: *TRM/EXT Command*



The TRM/EXT command allows lines to be trimmed or extended in several different ways. It is not restricted to use with line segments but also can be used with arcs.

PHOTO 4: *X-FORM Command*



The X-FORM command can be used to move, copy, scale, rotate, and mirror objects. This is particularly powerful with respect to CADKEY's three-dimensional capabilities.

entities, windowing, or selecting all objects currently displayed.

Creating a rectangular array is a two-step process, because the original set of entities must first be copied in one direction, then that row/column must be copied in the second direction. This apparent disadvantage is outweighed by the ability to create not only rectangular (two-dimensional) arrays, but also three-dimensional arrays, which requires only a third step. Arrays can be created along a vector defined during entry of the command.

X-FORM's SCALE option provides the ability to create a series of scaled copies of one or more entities, as well as moving and scaling an object at the same time. This allows rapid construction of pyramid objects; creation of sets of entities, which differ only in scale, from a single pattern; and rapid construction of perspective views by building an array of scaled, tapered elements using vanishing points as references.

The powerful 2D/ROT command of the X-FORM menu allows the construction not only of circular arrays of plane figures, but also of toroidal shapes in three dimensions (see figure 1). With a combination of primitive shapes, meshes, and the 2D/ROT command, the user can model toroids, spheres, domes, and arbitrary surfaces produced by rotating a shape around an arbitrarily inclined axis.

DETAILING

CADKEY's main menu includes a selection, DETAIL, to access a group of commands that provide the facilities for automatic dimensioning, cross hatching, and drawing notes—all essential elements of a production drafting system.

Automatic dimensioning is accomplished with the DIMENSN command. This command provides six options: HORIZTL, VERTICL, PARALEL, RADIUS, ANGULAR, and DIAMETR, which should cover all needs. These options prompt for the two witness line origins and the location of the dimension line, display a calculated dimension for verification, and draw the witness lines, dimension line, and dimension text if accepted. To speed the task of entering the witness line origins, the options display the position menu, providing the ability to snap to features of drawing entities—endpoints, centers, intersections.

As with most microcomputer-based CAD systems, the dimension is not associated with the drawing entity to which it refers. The user can dimension an entity and then move it without affecting the location of the dimension. However, the dimension is treated as a unit for some functions, particularly those editing functions contained in the DETAIL menu and submenus.

Other CAD systems may allow editing of dimension text, but not many others allow the dimension to be treated as a unit. In these systems, the fastest method often is to erase a dimension and reenter it rather than to modify it. On the other hand, erasing a dimension that is not treated as a unit may be difficult, because the elements of the dimension may be superimposed on other drawing entities.

In fact, CADKEY does not treat dimensions in the same manner as it treats other drawing entities. The editing commands that are used to move, copy, and join entities such as lines, circles, and arcs generally do not recognize dimensions. For example, X-FORM COPY WINDOW copies all the lines, circles, and arcs within the window, but not the dimensions. This method of operation is desirable because dimensions should not be copied, translated, and rotated blindly based on geometric relationships to other drawing entities. Dimensions should be disassociated from drawing entities unless they are actually attributes of the entities.

Various aspects of dimensions can be modified with the CHANGE command. It allows tolerances to be altered (which shortens the dimension text), text attributes (font, height, aspect ratio, and orientation) to be changed, text position to be adjusted, number of decimals displayed to be changed, and arrows to be switched from inside the witness lines to outside. All of these options contribute to CADKEY's ability to produce attractive dimensions.

The DIM SCL option of the CHANGE command provides another useful feature. This option leaves the dimension as is, with one exception—it scales the value of the dimension text. Therefore, a drawing of a part can be adapted to portray a scaled version simply by scaling the dimension values. Dimension values can also be changed with DIM VAL, which allows new dimension text to be entered.

While the CHANGE command modifies existing dimensions, the SET command modifies the parameters used in the entry of new dimensions. CADKEY produces dimensions in accordance

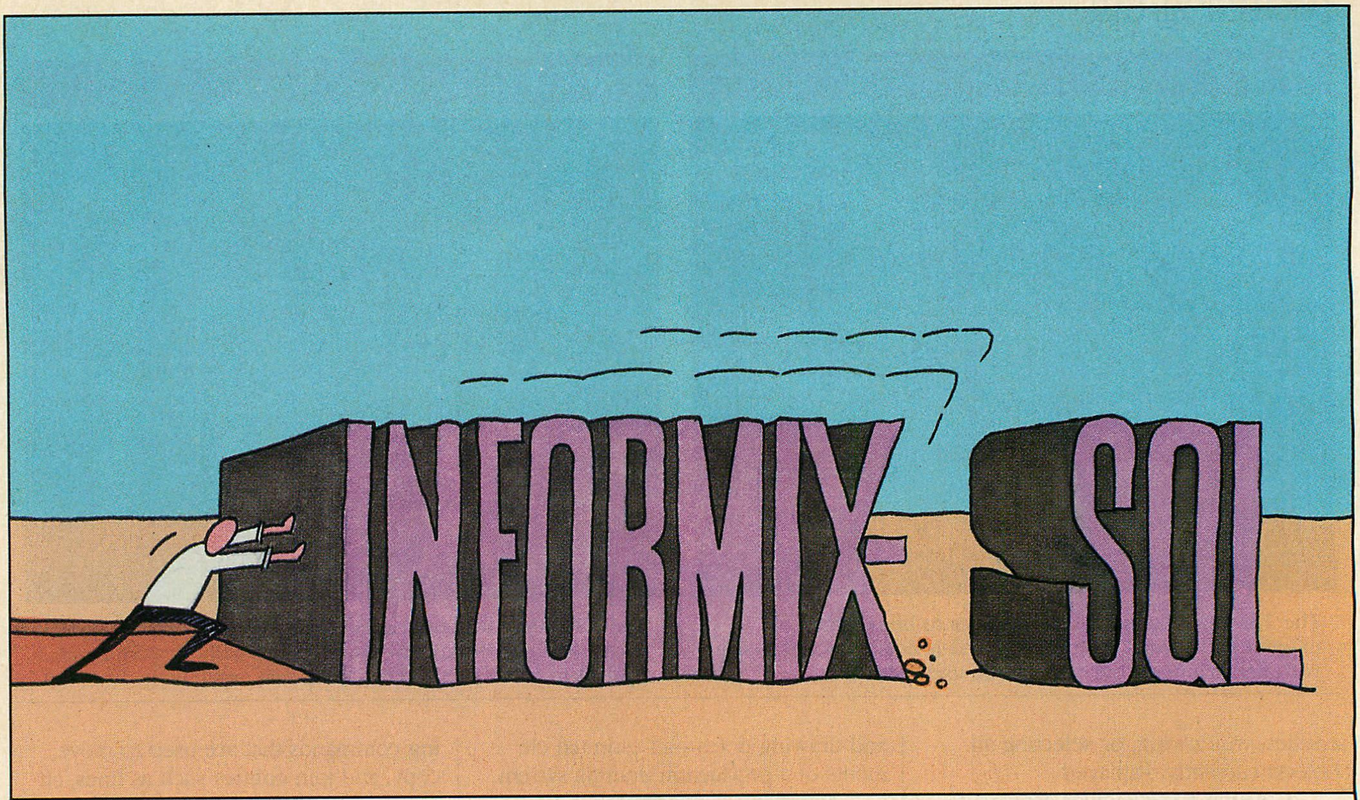
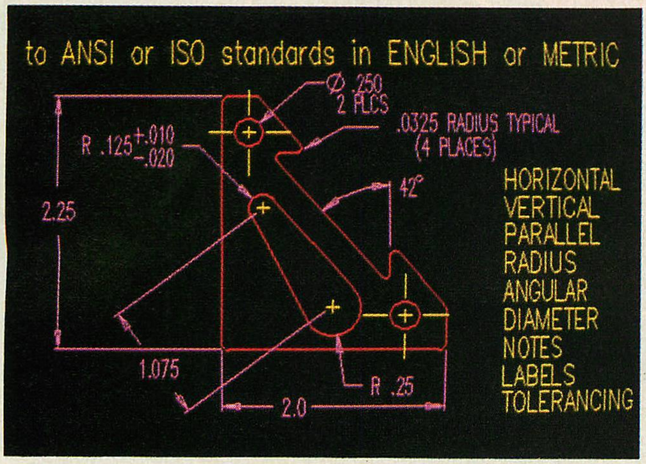
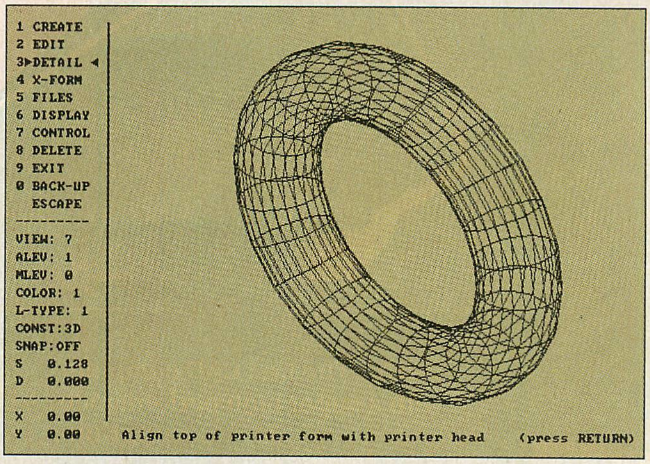

INFORMIX-SQL

PHOTO 5: *Dimension Standards*



The dimensions on a CADKEY drawing are quite versatile. The size and other attributes can be changed, although mixed units such as feet and inches are not possible.

FIGURE 1: *Printer Plot*



The Ctrl-Y command can be used to produce check plots on an Epson printer. The toroid shown above was created by using the transform command on a polygon shape.

with ANSI standards, but can be set to follow ISO standards using the SET command (see photo 5). SET provides extensive control of the parameters for the appearance of dimensions—arrow orientation, witness line visibility, decimals displayed, tolerance, display of leading and trailing zeroes, and text height, font, alignment, and aspect.

The UNITS option to the SET command allows units to be scaled to produce the correct numerical values for inches, millimeters, feet, centimeters, yards, meters, or user-defined values. The program does not suffix dimension values with unit symbols and does not support mixed formats such as feet and inches. SET UNITS does not rescale existing units, but sets a current scaling factor that applies only to all dimensions entered thereafter.

The UPDATE option allows certain attributes to be globally set to the current set of attributes. The affected attributes are height and font of notes and labels; and height, font, decimal places, dimension scale, trailing and leading zero display, tolerance mode, and tolerance values. The changes can be made to single or chained objects, objects in a window, and all objects displayed.

Dimensions are visible only in the view in which they are created. This means that a three-dimensional object can be dimensioned and plotted in all views without the distraction of edge-on dimensions being displayed in each one. If the object is dimensioned in the three orthographic views (isometric, axonometric, or user-defined), the views can be plotted from the same drawing without the distraction of dimensions being displayed in the various planes.

In addition to dimensioning, the DETAIL menu includes an X-HATCH option that automates the process of cross-hatching areas enclosed by lines and/or arcs (or circles). As with most paint, area fill, or hatching processes, the area must be enclosed or the hatching will leak out and fill the entire drawing.

Nested enclosed areas—called *islands* in the CADKEY manual—can be selected for hatching in one of three ways: alternating areas are hatched, all areas are hatched, or areas are hatched in an arbitrary sequence. The hatching sequence is determined by the selection of the hatching boundaries.

As with dimensions, hatching patterns are visible only in the view in which they are created; this eliminates distractions in other views. Hatching patterns are treated as single entities, but they are not associated with the hatching boundaries. Therefore, the patterns can be moved or rotated as units. Holes can be created in hatching patterns by selecting the outer boundary and a nested inner boundary, hatching the annular space, then deleting the entities making up the inner boundary.

CADKEY is supplied with seven hatching patterns—not an extensive library, but the essential patterns for mechanical drafting are available: brick, steel, copper, alloys, aluminum, rubber, and marble. A simple pattern of horizontal lines also can be selected. Hatching patterns can be created at specified scales and rotation angles.

Notes and labels are handled in CADKEY's DETAIL menu with the NOTE and LABEL options. The entities created by these two commands have two principal differences: labels have leaders and arrowheads and must be typed in from the keyboard, whereas notes do not have leaders and can be typed in or read from a disk file. Labels are limited to 256 characters including carriage returns. Notes are limited to 256 characters if typed in or to 1,024 characters if read from disk. The ability to read notes from a disk file greatly eases the use of key notes and boilerplates, because these files can be created externally with ordinary text editors; no conversion process is required prior to reading the file into the drawing.

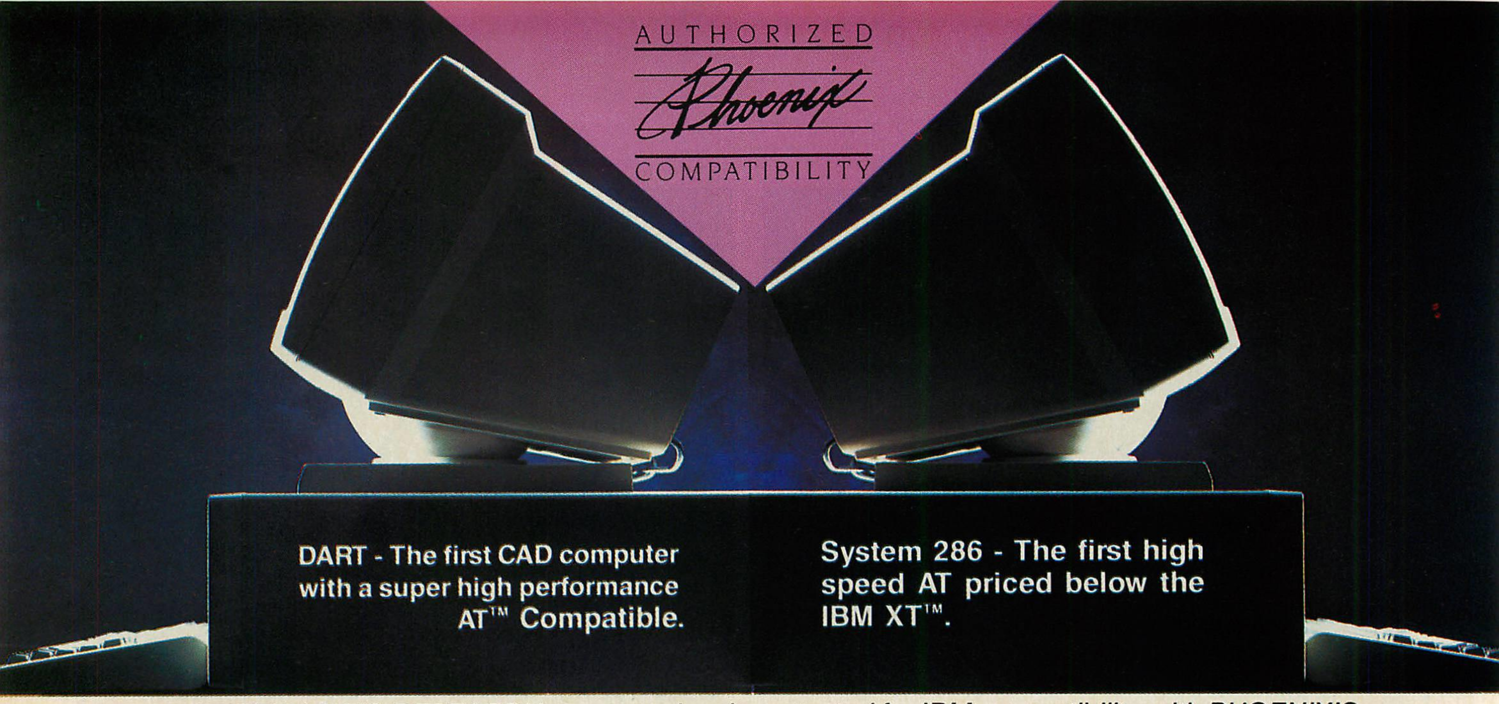
The program is furnished with three fonts, BOX, SLANT, and BOLD, which can be changed with the DETAIL SET TEXT command. Dimension and note fonts can be set independently. Changing the font partway through a drawing alters only subsequent entries.

PRINTING AND PLOTTING

Plotting in CADKEY is accomplished with a separate program, PLOTFAST. Within the main program, the FILES PLOT command is used to create a plot file that is used by the plotting program as plotting data.

The FILES PLOT command has two options: LIST and SAVE. LIST displays the plot files in a selected subdirectory. Each entry includes the file name, file size, and date and time of creation. If the entire list does not fit on one screen, the program pauses until the user presses the Enter key.

The SAVE option prompts for a file name and creates a plot file from the current drawing database. If the current plot directory already contains a plot file with the specified name, the old file is overwritten, with no prompt.



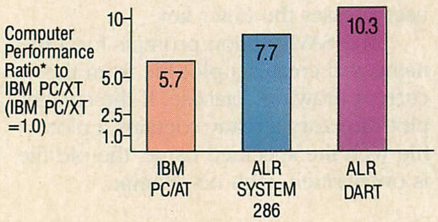
The ADVANCED LOGIC RESEARCH computer has been tested for IBM compatibility with PHOENIX'S compatibility test suite using the PHOENIX AT-COMPATIBLE ROM BIOS, and the PC-COMPATIBLE ROM BIOS.

WE DON'T THINK YOUR PERFORMANCE SHOULD BE LIMITED BY THE 'STANDARD'

At ALR we don't think your performance should be limited by the 'Standard'. So preoccupied are today's computer makers with meeting the 'Standard', they seem to have overlooked the opportunity to do something much more worthwhile.

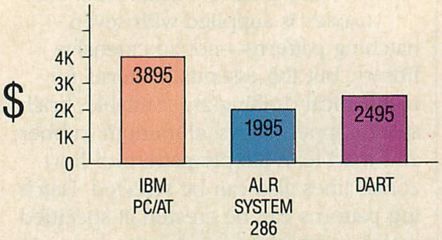
Namely, the concept of improvement beyond that 'Standard' acceptable level.

Examine the DART. The first CAD computer with a super high performance AT™ Compatible. The DART system is powered by a 10 MHz 80286 cpu with support for an 8 MHz 80287 math co-processor, when compared to the IBM® AT's 6 MHz cpu and 4 MHz co-processor, the DART system will process your next CAD design in almost half the time, with DIRECT ACCESS RESPONSE TIME (DART).



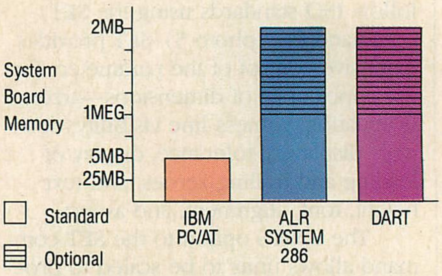
*Based on the Norton Utility SI-System Information Command.

Explore the opportunity of the System 286 - The first high speed AT below the price of an IBM XT™. The low cost of this system did not reduce the performance, with the cpu speed of 8 MHz and co-processor speed of 5 MHz. This System 286 is ready to calculate spread sheets and sort data base in a flash!

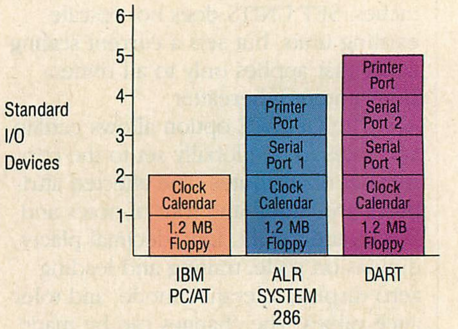


LIST PRICES AS OF APRIL 1986

The concept of having enough memory is misleading. It seems yesterday's 64K of memory was enough. Today, computers require at least 512K in order to run popular programs such as Symphony™ or Framework™. The result? ALR System 286 and DART system both have four times the memory capability built into its 'motherboard' so you don't have to pay for it tomorrow.



Option I/O device? That usually implies you need it and it costs more. ALR believes a computer needs to interface with at least a printer and in most cases modems, plotters and other computers, so we design our systems to include more standard I/O.



High performance and more features usually results in a more expensive system. However, the ALR System 286 and DART system does not match the 'Standard' price and this is one category we're glad to be considered 'sub-standard'.

Let one of our sales engineers or dealers show you the ALR System 286, DART system, Turbo XT and AT Hirise IBM compatible systems that are designed to limit your cost not your standard of performance.

CIRCLE NO. 199 ON READER SERVICE CARD

ADVANCED LOGIC RESEARCH, INC.
10 CHRYSLER
IRVINE, CA 92714
(714) 581-6770
ALR IN CANADA
416-229-6477

ALR

ENGINEERING ADVANCED COMPUTERS

The file created with the FILES PLOT SAVE command is a text file in the UNIPLOT file format. This is an ASCII file containing integer and floating-point data used by the plotting program. Plot files can be edited with text editors or word processing programs to change attributes and plotting data.

In the standard configuration, plotting is accomplished by running the PLOTFAST program, which displays a menu and prompts for the name of the file to be plotted. The parameters offered include plotter type, media size, plotting scale, *X* and *Y* offsets, serial port, pen velocity, and baud rate.

PLOTFAST supports several plotter series, including the Hewlett-Packard HP7400, HP7500, and 758x series, and the Houston Instrument DM/PL and other plotters that use the same protocol, such as Nicolet Zeta and Roland.

Once the plotting parameters are entered, three options are available: PLOT, RESTART, and EXIT. PLOT is the default response, and pressing Enter or Esc when it is displayed starts the plotting process. As the program sends data to the plotter, it displays an item count to indicate the plotting progress as well as the current pen number.

The program supports both multi-pen and single-pen plotters for multi-pen plots. The Houston Instrument single-pen plotters pause for pen changes, so the pen number display is useful. PLOTFAST searches the drawing for all entities to be drawn with pen number 1, then pen number 2, etc. to minimize pen changes. The ability to plot with multiple pens, even on a single pen plotter, is noteworthy. CADKEY does not support line widths except on the screen display, so pens of different widths must be used to plot lines of different weights.

The CADKEY literature recommends that plotting be done from a separate computer than the one used for the CAD workstation. This is a valid point because the speed of the plotting process is determined by the speed at which the plotter can accept data, not by the speed of the computer or plotting software. The PLOTFAST program requires 128KB of RAM, but does not require a graphics board or the software interface module for operation.

Screen dumps to a printer that accepts Epson-type control codes can be generated from within CADKEY. The program prints a copy of the screen—menus and all—scaled to fit an 8½-by-11-inch page, with the plot rotated 90 degrees (see figure 1). If the printer is off-line when the print job is requested, the user is sent to DOS with a divide overflow error. Printer plots are useful for cursory checks or illustrations.

SYMBOL LIBRARIES

A feature of a microcomputer-based CAD system that is essential for production drafting is the ability to define and use symbol libraries. Following the scheme established by the name of the basic drawing file—*part file*—CADKEY calls a symbol library file a *pattern file*. Only one part file can be active at a time, but pattern files can be merged with a current part file, or drawing. Pattern files can be inserted into the current part file at arbitrary locations, then scaled and rotated.

The FILES command of the main menu provides the facilities for creating and retrieving pattern files. The command FILES PATTERN CREATE creates a pattern file. CADKEY prompts the user to select the entities that will make up the pattern. Four methods of entity selection are available—SINGLE, CHAIN, WINDOW, and ALL DSP. After the entities are selected, the program prompts for a base point, which will be used to reference the insertion of pattern instances. Finally, the user is prompted for the pattern file name.

The process of creating a pattern file does not remove the selected entities from the screen and always creates a disk file.

The program places the pattern file in a directory, which is specified in the pattern file name as a path, or in the default pattern file subdirectory, which is defined in the configuration process. The path and file name string is limited to 40 characters.

The process of creating a pattern file does not remove the selected entities from the screen and always creates a disk file. This ensures that all defined patterns are available to all part files.

Patterns are inserted in part files with the FILES PATTERN RETRIEVE (or LST/RTV) commands. The program prompts for a pattern file name, an insertion point, a scale factor, and a rotation angle. The pattern files available in any directory can be displayed and selections made with the pointing device.

Patterns are not single entities. When a pattern is retrieved and placed into a CADKEY part file, it is not referenced. Instead, it is copied into the part file as a group of separate entities. Each entity can be moved, rotated, scaled, deleted, or edited as can any other entity. Therefore, if a complex pattern is inserted incorrectly, repositioning it—or even deleting it—can be difficult.

MACRO FACILITY

CADKEY provides a macro facility in its CADL (CADKEY Advanced Design Language) files. A CADL file is an ASCII text file—actually a database file consisting of variable length records. Each record is a key word—the name of an entity or a command, with the required parameters or arguments. CADL files can describe a number of entities, the commands to manipulate the display, or both, and they can be executed from the FILES menu, with the drawing adjusted accordingly. CADL files are created in one of two ways: they are extracted from the database or generated with a text editor. A combination of the two methods is possible.

Drawing entities in a CADL file can be located in either world or view coordinates. The set of data primitives includes view rotation matrices. The drawing entities allowed are arcs, circles, lines, points, and text strings. Lines and points can be entered in world or view coordinates. However, an arc or circle cannot be specified in the same manner. A circle, for example, requires a scalar value for the radius and the coordinates of three points to determine the plane in which the circle lies.

The view record of a CADL file sets the plane in which circles and arcs are contained before they appear in the file. A view record needs to appear only when the view changes, so several circles and arcs may follow the record.

The CADL file produced by the FILES CADL OUTPUT sequence provides a static description of a drawing. The file also can contain commands that simulate series of keystrokes entered from the keyboard. The only commands supported are those for display control. The ability to put program instructions in the CADL file is not available in the current release of CADKEY, but Micro Control plans to offer this in its next release, scheduled for release in May.

As currently implemented, CADL files can be used for two purposes: exporting coordinate data to external programs for nongraphic processing and importing data from external programs for the display of design data. For ex-

ample, the user could draw a structural framework in three dimensions and export the coordinate data of all the members via a CADL file. An external structural design program could scan the CADL file to determine nodes and connectivity of the framework, prompt the designer for loads and joint types, and size the structural members. Then, the structural design program could produce another CADL file, containing not only the original geometry, but also text to indicate member sizes and joint types. The CADL file produced by the structural program can then be executed, saved as a part file, and plotted.

At the present time, CADL files cannot perform the same functions as do macro facilities in some CAD programs. When the program instructions are implemented, however, a true macro facility should be available.

DRAWING TRANSLATION

CADKEY's developers support two recognized standards for the interchange of drawing data, in addition to providing a readable form of their own database. These facilities allow for the free interchange of drawing data from CADKEY to other CAD programs—both microcomputer-based and larger systems.

DXF. The DXF file is the Drawing Interchange Format that was originated by CalComp's AutoCAD. Although CADKEY and AutoCAD are competitors, CADKEY's creators have wisely chosen to support the DXF file method of drawing data interchange. The CADKEY program provides a means to save part files in the form of DXF files and to load or retrieve data from DXF files.

Saving and loading DXF files are supported to different degrees. CADKEY can load a DXF file containing the following entries: ARC, BLOCK, CIRCLE, ENDBLK, ENDSEC, INSERT, LINE, POINT, POLYLINE, SEQEND, SOLID, TEXT, TRACE, and VERTEX. This assortment of entities allows most of a DXF file created by AutoCAD or another program supporting DXF files to be imported correctly. CADKEY generally imports the DXF entities into the current level and assigns the current color, line type, and line width. (Line width is currently associated only with the display.)

Some entities cannot be imported correctly, because CADKEY and AutoCAD do not describe them in the same manner. For example, AutoCAD allows for blocks to be inserted with different *X* and *Y* scale factors, providing control of aspect ratio, whereas CADKEY's patterns are inserted with a single scale factor that applies equally to all three dimensions (AutoCAD has the same restriction with respect to the insertion of blocks that are preceded by the * character). Arcs and circles contained in DXF block references with unequal *X* and *Y* scale factors are not imported, because CADKEY has no way to represent objects with different *X* and *Y* scale factors and no primitive data type for ellipses. Lines and points in the same block references can be transformed so that they are represented correctly, and, therefore, they can be imported.

Presumably because AutoCAD has only a single coordinate system and is basically an enhanced two-dimensional system, CADKEY considers all DXF coordinate data to be view coordinates, rather than world coordinates. (AutoCAD's data structure does not provide for objects in planes inclined with respect to the *X-Y* plane, only for planes parallel to the *X-Y* plane, so it has no straightforward way to interpret DXF coordinates as world coordinates.) However, the user should be able to assemble a three-dimensional CADKEY model by importing enough orthographic and auxiliary views.

CADKEY's creators have wisely chosen to support the DXF file method of drawing data interchange.

DXF files created with CADKEY are limited subsets of the complete DXF file specification, again, because CADKEY and AutoCAD do not have completely parallel data structures. Three options are available for selecting the entities that will be placed in the DXF file: SELECT, LEVEL, and ALL ENT. SELECT prompts for the selection of some or all of the entities visible on the screen. Selection of visible entities has a few inconsistencies, although the scheme is workable once understood. Selecting single entities works as expected unless several entities are stacked, in which case CADKEY indicates that an entity has been found, but nothing is output. Selecting entities with a WINDOW or ALL DSP option exports stacked lines, lines perpendicular to the viewing plane, and circles and arcs in planes that are parallel to the viewing plane. The coordinates of the lines and points, regardless of the view, are transferred with world *X-Y* coordinates, and the coordinates of circles and arcs in the current view plane are transferred with view coordinates—as if they were world coordinates.

Because circles and arcs are specified with centers that have *X-Y* coordinates and radii that have a scalar value, the center of the circle or arc will be in a predictable place when the drawing is transformed into the DXF file. However, the value for the radius is taken regardless of the displayed view used. This means that the representation of the circle will be wrong in the transformed file unless the circle or arc was in the same plane as the transformed view.

One last note on the use of DXF files: CADKEY creates a minimal DXF file, including only the data that can be associated with the POINT, LINE, ARC, and NOTE (text) entities. The header, table, and block sections of the DXF file are empty in the output file. Only the entities section contains entries.

IGES. Whereas the DXF file is the industry-accepted method of interchanging drawings from one microcomputer-based CAD system to another, micro-to-mainframe exchange is normally accomplished with the IGES standard. The CADKEY package supplies an IGES translator as an extra cost option.

GEOMETRIC ANALYSIS

CADKEY provides a group of three options under the CONTROL VERIFY command that should prove useful in design work at the keyboard: PERIM, AREA/CN, and MOMENT.

PERIM calculates the perimeter of circles, arcs, and arbitrary line strings. It has two options: CIRCLE prompts for the selection of a circle or arc, then displays the perimeter of the circle projected on the current view plane, as well as the actual perimeter; GENERAL displays the selection menu, which as with other commands allows the selection of isolated objects, chains of objects, all objects in a window, or all objects currently displayed. Again, the program displays the perimeter projected on the current view plane and the actual length of the selected objects.

AREA/CN calculates the area and centroid of a closed shape or of the segment of a circle formed by adding a temporary chord to an arc. The command provides three options: CIRCLE, POLYGON, and GENERAL. POLYGON and GENERAL require the selection of a closed shape, POLYGON for lines only, and GENERAL for lines and arcs. The entities that form the boundary must meet precisely. When a valid selection has been made, the program displays the area on the prompt line. Pressing

the Enter key displays the *X* and *Y* coordinates of the centroid, or center of gravity, which is calculated by dividing the area into a series of rectangles, referred to as *panels*.

GENERAL prompts for the panel width, which affects the accuracy of the area and centroid calculations. The narrower the panel, the more accurate the results. The command actually draws the panels on the screen as it calculates the area, in some cases obliterating the display. A redraw clears the screen of the panel lines and other markers.

The MOMENT option of the CONTROL VERIFY command calculates the moment of inertia of the shape about a specified axis and has the same general format as does the AREA/CN option. However, in addition to prompting for the shape, and panel width in the GENERAL case, the program prompts for a reference axis about which the moment is taken. The axis is indicated as a point, using the position menu, and is a temporary vector passing through the specified point, perpendicular to the view plane. Again, the program prompts for a panel width. The calculation is not instantaneous, but usually takes only a few seconds. Using larger panels widths speeds the calculation.

Aside from the restriction imposed by keeping its entire drawing database in memory, CADKEY has few limitations. The program uses 33 overlays, which produce a fair amount of disk activity. Virtually every menu change requires a disk access. If menu selections are made with the function keys, response is fast and no keystrokes are lost, even though the disk access may delay the screen output. With a pointing device, the user can enter commands faster than the program accepts them, due to the slight delays of disk activity and rewriting the menu window.

As expected, the program operated considerably faster on an AT than on a PC. Loading the program on a Heathkit H-200 AT compatible required approximately 13 seconds, whereas a PC with add-on hard disk and V20 microprocessor took approximately 22 seconds. View changes and autoscaling took several seconds, but varied depending upon the number of entities in the drawing. The speed of redraws were on a par with other CAD packages.

The program was tested on a Priam SharedSpace (formerly ClusterTower) external hard-disk unit, connected to an AT and a PC. Loading the program into the AT required 17 seconds, compared to 14 seconds for loading the program from the AT's internal 30MB (type 20) disk drive. In other respects, the speed of the program seemed identical when run from the external or the internal disk drive. Loading the program from the SharedSpace unit to the PC required approximately 25 seconds, again only slightly longer than loading it from the internal disk drive. The primary advantage of using the external unit is that pattern libraries can be stored in a single location and made available to several workstations, thereby reducing redundant storage, eliminating the possibility of retrieving an out-of-date pattern file, and simplifying the task of pattern file management.

DESIGN EXTENSIONS

In its standard form, CADKEY is a CAD program—it performs basic drafting tasks and is not specific to any particular design discipline. However, the vendor apparently intends for the system to be the basic component of more advanced CADD systems.

CADKEY is a powerful 2-D/3-D production drafting system, although its lack of certain features may limit its appeal to some designers—for example, the lack of feet and inches units displays is a drawback to architects and interior designers. Furthermore, it is not a 3-D modeling system in the usual sense. Its set of data primitives includes no cubes, spheres, or cylinders. Nor does it create perspective views automatically nor remove hidden lines. However, for many designers, CADKEY can serve as a single tool for creating orthographic drawings and wireframe models of three-dimensional objects and for constructing perspective views.

> *CADKEY seems best suited for mechanical design and drafting in its current configuration, but should be useful in many fields.*

CADKEY also provides several geometric analysis tools useful to designers, particularly in the area of machine design. This facility, if expanded, could set CADKEY apart from many CAD programs. The program seems best suited for mechanical design and drafting in its current configuration, but should be useful in many fields, including technical illustration, especially with its ability to read text into a drawing from a disk file prepared with a word processor.

CADKEY documentation is substantial. The manual is contained in an 8½-by-11-inch loose-leaf binder that is more than an inch thick and filled to capacity. It is organized with tabs corresponding roughly to the main menu selections. Each main menu section tab contains a chart illustrating the various paths from the main menu selection to the bottom of the menu structure.

CADKEY provides a user interface that most designers should understand. The paradigm of viewing planes that can be rotated and translated arbitrarily is strange at first, but, once grasped, it seems the natural way to work.

All menus are furnished as text files that the main program reads into memory. Although the CADKEY manual does not provide details on customizing these menus, it does note that they can be altered within certain limitations. Prompts are also furnished as text files, so they too can be altered for custom versions of the program.

By providing for both DXF and IGES file translation, Micro Control Systems has made CADKEY drawing files available to design programs written to industry-standard formats. This is an important point in the selection of a CAD program. Although the CADL facility may encourage third-party vendors to extend the program and may be used for any design extensions provided by Micro Control Systems, it is unlikely to supplant the DXF or IGES formats. Third-party developers will follow the formats that promise the largest markets—DXF and IGES.

The CADL facility must be considered a preview of future versions of the program. Currently, it is a rudimentary facility for import and export of graphic data for use with other design programs. When the command facility promised in the manual is added (BASIC-like statements), the user may be able to extend the program with interactive macros and perform design tasks within CADKEY.

CADKEY: $2,695
Micro Control Systems
27 Hartford Turnpike
Vernon, CT 06066
203/647-0220
CIRCLE 338 ON READER SERVICE CARD

Victor E. Wright is the manager of process engineering at Luckett & Farley, located in Louisville, Kentucky. This article is his fourth in a series of reviews of micro-based computer-aided drafting systems.

Instant Screens

By using the horizontal and vertical retrace periods and swapping the visual and active page, users can achieve almost instantaneous display changes without snow for text pages on a CGA.

AUGIE HANSEN

Since its introduction in 1981, the IBM PC has been offered with the choice of monochrome or color/graphics display systems. Initially, the majority of systems were configured for monochrome only because it was the less expensive alternative. The situation has changed dramatically in just a few years. Now surveys and estimates of new sales show a more even distribution of monochrome and color/graphics displays. Of course, both may be installed on the same machine, although most programs are written to display output to only one at a time.

DOS provides a set of routines that establish the video mode, write data to and read data from display memory, and perform a variety of related tasks. IBM recommends that program designers use the routines provided with DOS and in ROM BIOS to achieve portability of programs both to new versions of DOS and to new members of the PC family. In IBM terminology, programs that adhere to the DOS/BIOS interfaces are "well behaved." The penalty for good behavior is a slower execution for screen-intensive applications because of the high overhead associated with the DOS and BIOS calls.

For many programs, the portable approach is the way to go. But the PC public has an appetite for sizzle, which sells as much software as it does steak. A quick look at the top-selling software products bears this out: programs that have snappy screen displays (assuming requisite good performance otherwise) have a better chance in the marketplace

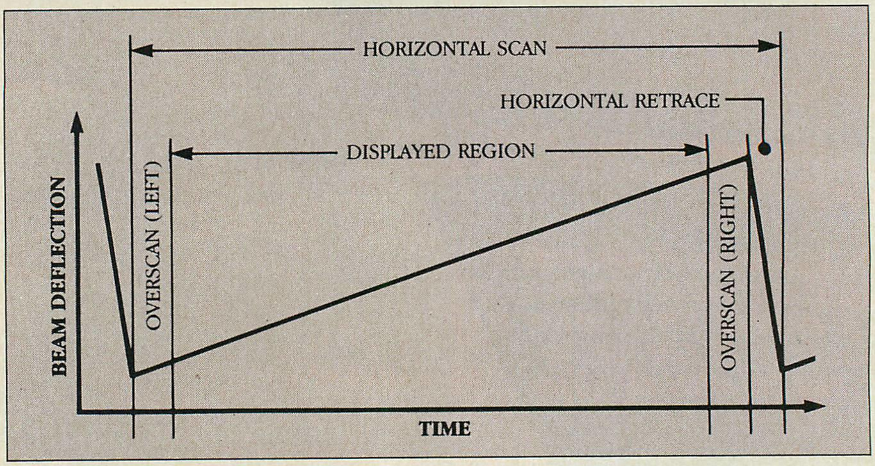
ILLUSTRATION • ANDY LEVINE PHOTOGRAPH • JOHN LEI

than programs that do not. For example, an editor that takes a second or two to scroll the screen is going to capture less attention than one that does basically the same job but scrolls the screen in one-tenth of a second or less. Programmers can be an impatient lot, sometimes willing to pass up otherwise excellent editors, even free ones, because they operate slowly when updating the screen.

An inherently faster but decidedly less portable way of updating the screen is to write directly to its associated memory. A convenient way of managing the screen is to keep off-screen buffers (one or more) in the program's data space and to use a block-copy routine that quickly shuffles a memory image to physical display memory. For the monochrome display adapter, this strategy causes no problem. It also works for the IBM Color Graphics Adapter (CGA) in any of the graphics modes. However, in either 40- or 80-column text mode, the original CGA is a problem because, unlike the monochrome adapter, the CGA exhibits visible interference when a program tries to access display memory while the screen is being updated.

This interference, an irritating pulsing of short lines covering portions of the screen, is referred to as *snow*. Several methods of avoiding snow are available. One way is to synchronize the display accesses during reading and writing operations with the time periods within a display refresh cycle that are considered safe. The safe times are the horizontal and vertical retrace periods of each displayed frame. Another involves blanking (turning off) the raster scan while the display memory is being written to. Each approach carries advantages and disadvantages.

FIGURE 1: *Raster Scan*



Any transmission of data during the period that the beam is writing to the screen will cause visible interference. Transmissions must take place during retrace.

DISPLAY ADAPTER BASICS

The screen interface package can be best appreciated by first understanding why the CGA retrace periods are the only safe times for display memory accesses. (This discussion does not apply to the IBM Enhanced Graphics Adapter because it uses faster memory devices and additional logic that precludes problems with simultaneous access by the CPU and video refresh circuitry.)

The memory on the display adapter is placed within the address space of the central processor. The CGA memory begins at B800H and extends upward for 16KB, enough for one high-resolution graphics screen (128,000 pixels), or four screen pages in 80-column text mode. Text mode is the focus here.

Figure 1 shows the horizontal sweep signal, the signal within the device that is responsible for the horizontal deflection of the electron beam that paints the screen. The figure depicts one horizontal scan period. The dependent axis (up) is the beam deflection as a function of time shown along the independent axis (across). On a computer display, the screen image is *underscanned*—the image is completely visible within the normal viewing area. The result is a framed picture with a border. (In contrast, television sets use *overscanning* to make the image bleed, that is, leave no border.) The beam is turned off completely (blanked) during retrace to avoid leaving unwanted residue on the screen.

The IBM display is noninterlaced, which means 262.5 horizontal scans occur per frame (one full screen image), and 60 frames occur per second. With 15,750 horizontal scans per second, each takes about 63.4 microseconds. Only a small portion of a single scan, typically 20 percent or less, is allocated to the horizontal retrace (one of the safe times for display memory accesses) as shown in figure 1. One byte of data can be copied safely to or from display memory during this period.

Programs that copy entire words (16 bits) in horizontal retrace may get by on some machines but fail on others. The original CGA board on one tested PC showed annoying jitter and interference along the left side of the display area when running programs that try to copy words instead of bytes. The behavior appears to be temperature sensitive, indicating that transferring more than a byte per retrace period leaves too little of a safety margin.

The horizontal sweep signal moves the electron beam from side to side. If it were the only sweep signal affecting the beam, nothing but a single straight line would appear on the display surface. Another kind of sweep signal is needed to move the beam up and down the face of the tube. This is the vertical sweep and it has the same basic sawtooth shape as the horizontal sweep, but a slower rate of change. At minimum deflection, the beam is at the top of the screen; it is moved toward the bottom as the deflection increases.

A vertical retrace period takes place during the vertical synchronization pulse period at the end of each frame. During this time, the electron beam is blanked and moved from the lower right corner of the screen back to the upper left corner. This period is long enough—a little more than one millisecond—to permit a block of 250 data words (character and attribute pairs) to be transferred to or from display memory without interference.

VIDEO INTERFACE

A few important considerations affect the design of a program that will interact tightly with the display system. Because of the choices available in the design of a video application, no two designers are likely to accomplish the same task in the same way. The following is just one method.

The first decision made in this design was to use a buffered screen interface; that is, an image of what is to be sent to the display is assembled first in the program's own data space. When complete, the image is copied as quickly as possible to display memory via a block-copy routine where it is periodi-

cally repainted on the screen by the raster-scan logic. *Raster* is the pattern of tightly-spaced horizontal lines that produce the displayed image on a cathode ray tube (CRT).

Many programs use an unbuffered approach, which is adequate for most purposes. Characters are written into display memory via DOS and BIOS routines, but no memory image is retained by the applications program.

Most advanced microcomputer users want an instant response in screen update; for example, they expect a command selection from a menu to be displayed as soon as the key is released. A few tenths of a second or less is a good rule of thumb. Such a response is attainable with a modest amount of programming. As noted above, however, portability to machines other than the IBM PC may be sacrificed; special versions of the screen-interface programs may be required.

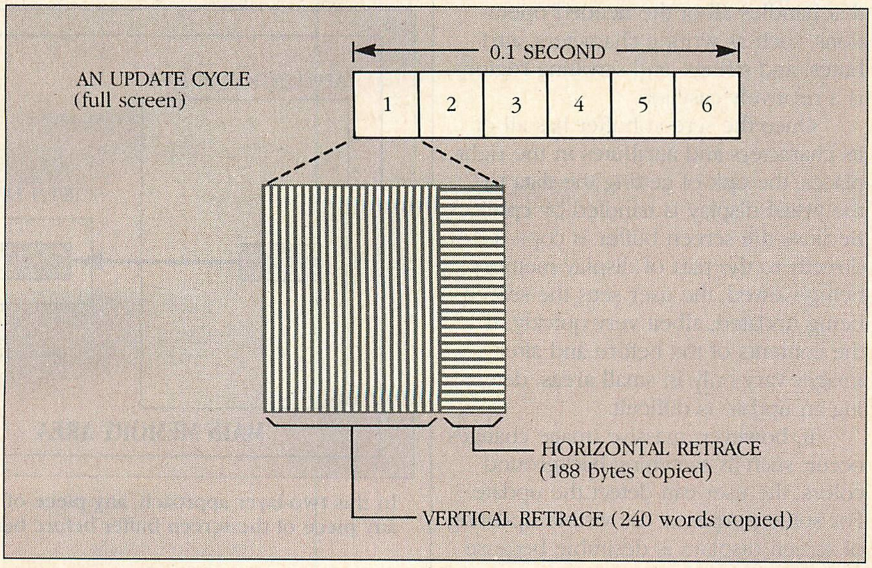
The routines described in this article assume that programs calling them already have done an equipment inventory and set up the display system in the 80-column text mode.

Available methods of synchronization depend on the use of the status register at I/O address 3DAH. This is a read-only register on the CGA that has two bits of interest to the block-copy routine: when bit 0 is high, it indicates that a horizontal retrace is in progress; when bit 3 is high, it indicates that a vertical retrace is in progress.

As noted, single bytes can be written during horizontal retrace periods, and large blocks of data words can be written during vertical retrace periods. To compress the time needed to transfer a screen image, the two approaches are merged using a combination of the horizontal and vertical retrace periods (this is explained with the block-copy routine that follows).

The mode control register, a write-only register at I/O address 3D8H on the CGA, has a bit that may be reset (to 0) to disable video. Bit 3 must be set to 1 to turn on the beam that paints the screen. Turning off the beam is an effective way to prevent snow. However, the beam cannot be left off for more than about three character rows of data before flicker becomes apparent. The BIOS video scroll routines use this technique and are unpleasant to view if the background color is not black.

FIGURE 2: *Block Transfers*



One byte of data can be transferred during each horizontal retrace and 240 words can be transferred during a vertical retrace, all without causing snow.

BLOCK COPY

The instant screen method described here uses a memory buffer that holds the same amount of data as one display page on the standard CGA. The block-copy routine, CPBLK.ASM (listing 1), copies the contents of the memory buffer to display memory only during safe times. The memory buffer has 4,000 bytes: 2,000 are for characters (25 rows by 80 columns) and 2,000 hold the attributes associated with each character. Display memory has 4,096 bytes per page (four pages in 80-column mode), but the last 96 bytes of each page are unused by most programs.

The source code for **cpblk** is coded with conditional pseudo operators and, therefore, must be assembled using a macro assembler that recognizes Microsoft assembler pseudo operators. Alternatively, the conditional assembly lines can be removed and the file edited for customization to individual compilers used for the C language parts of the package.

An image is copied from application memory to display memory in blocks. Six blocks may be copied in one-tenth of a second. Each block has two parts, as shown in figure 2. A full 240 words are copied during the vertical retrace period in one string move operation. Another 94 words are copied as single bytes, one per horizontal retrace period. These values are based on both calculations and empirical measurements and are conservative enough to work with a worst-case display adapter without causing visible interference. (The color display system tested experienced very noticeable interference when an attempt was made to copy more than 248 words during vertical retrace and more than a single byte during a horizontal retrace period.)

Interrupts are turned off only during the critical time when **cpblk** is waiting for the start of a horizontal retrace period. This is done because even the slightest delay (by a keyboard or clock interrupt, for example) could cause a display memory write at the wrong time, resulting in snow. The vertical interval is relatively long and it has enough of a safety factor for the number of words being transferred that interrupts are left on. If interrupts were turned off during the vertical retrace period, problems would result for other programs, such as the clock update routine and resident utilities.

DOUBLE BUFFERING

To produce the snappiest performance from this display interface technique, programs should use an in-memory screen buffer that is updated out of the user's view, then copied to display memory as quickly as possible. A method of achieving nearly instant CGA updates is shown in figure 3. The technique is called *double buffering* because two levels of buffers are maintained in the application program. A two-step process is used to form a composite image in a screen buffer before it is copied to physical display memory.

Application buffers may be any size and are usually thought of as rectangular. Any piece of them can be mapped to any piece of the screen buffer as required. This technique permits windows for help frames, menus, and so on

to be overlaid easily onto another image. Writing a set of library functions that handles all of the needed operations, such as writing characters, attributes, and strings, and scrolling regions, is a relatively easy job.

Once the screen buffer has all of its characters and attributes in the right places, the task of getting the data to the visual display is handled by **cpblk**. Because the screen buffer is copied directly to the part of display memory being viewed, the user sees the screen being updated, albeit very quickly. If the contents of the before and after images vary only in small areas, detecting an update is difficult.

If, however, massive image changes occur, such as switching background colors, the user can detect the update. For some purposes, the visible updating of screen displays is desirable because it may reassure the user that the program is working. For a program that is meant to produce smooth animation effects, this is not sufficient.

A DEMONSTRATION

Listing 2 is the C source code for a test driver program called ST (for screen test). The **#ifdefs** are provided for Computer Innovations C 2.3A, Microsoft C 3.0, and Lattice C 3.0G. Other C compilers may require different code to obtain the data segment register value needed for the block transfer. ST dynamically creates five screen buffers in main memory and loads them with known values. The character used for each of the screen buffers (0 through 4) is its number. Differing color attributes show the effect of massive color and intensity changes.

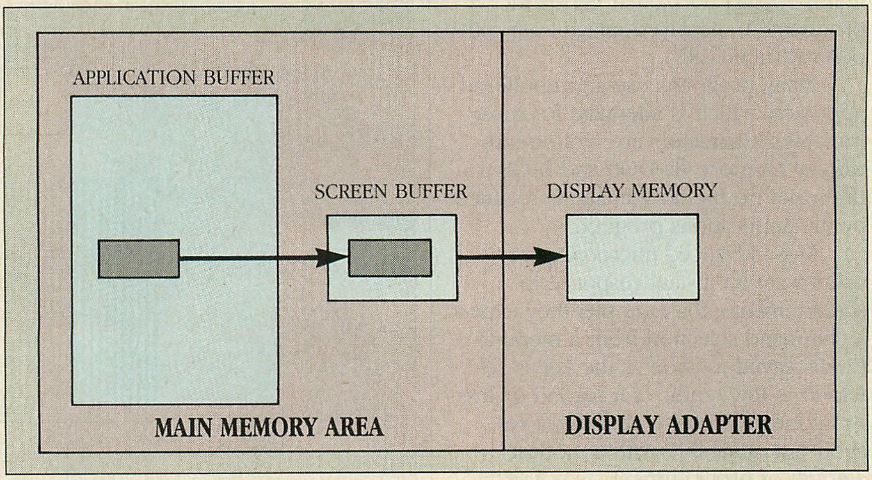
The **wait_ch** function (listing 3) calls DOS function 8H, which returns the next available character in the keyboard buffer. The function waits for a key to be pressed if nothing is ready. If the returned value from the DOS call is a null byte, **wait_ch** reads another, which is the value of an extended code from the keyboard. The driver program displays a help message for any key pressed except 0-4, Esc (the quit command), and Ctrl-Break (abort).

The driver program is using a trick to produce the appearance of instant screen displays. About one-tenth of a second is required to copy a screen buffer in the application data space to the display adapter using **cpblk**. Although this is fast in comparison to other methods that also curb video interference, it is still far from instantaneous. The trick is called *page flipping*, a method made possible by the fact that the CGA has enough display memory to hold multiple screen pages simultaneously (see figure 4).

Page flipping depends on having a means of telling the display system to view one page of display memory while the application is writing to another. The ROM BIOS video interrupt has a function (5) that sets the visual page. User should ignore the *IBM Technical Reference* statement that the function sets the active page. To be consistent with the way BASIC describes video pages, the active page should be the one being written, and the visual page should be the one being viewed. Frequently, they are the same page.

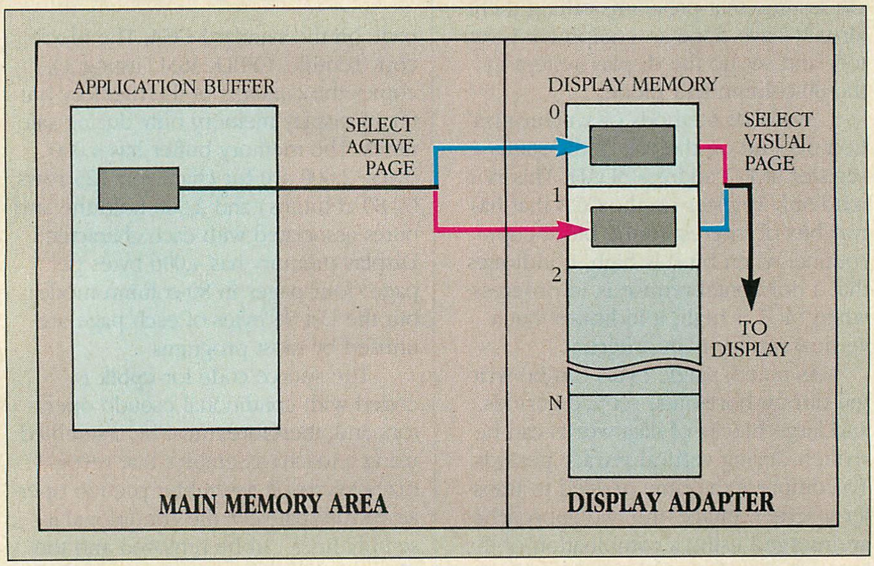
Notice that the **fprintf** standard library function writes to standard error, which appears on the visual page. **Cpblk**, however, is directed to write to the active page, which is effectively hidden from view. When the active page has been fully written, it is revealed to the user by flipping the pages. The function **swap_pg** exchanges the values of the **Apage** and **Vpage** variables, then calls **setpage** (listing 4) to switch to the new visual page. The effect is an instant update from the user's perspective. A short delay takes place while the active page is being updated before the page swap, but the user will not be able to detect it: the screen is repainted in

FIGURE 3: *Double Buffering*



In this two-layer approach, any piece of the application buffer can be written to any piece of the screen buffer before being transferred into the display memory.

FIGURE 4: *Page Flipping*



The display adapter is capable of storing four complete screens of text. The monitor displays the visual page as the active page is being updated. Then the visual page and the active page are swapped creating an instant screen change.

PC Magazine Product of the Year

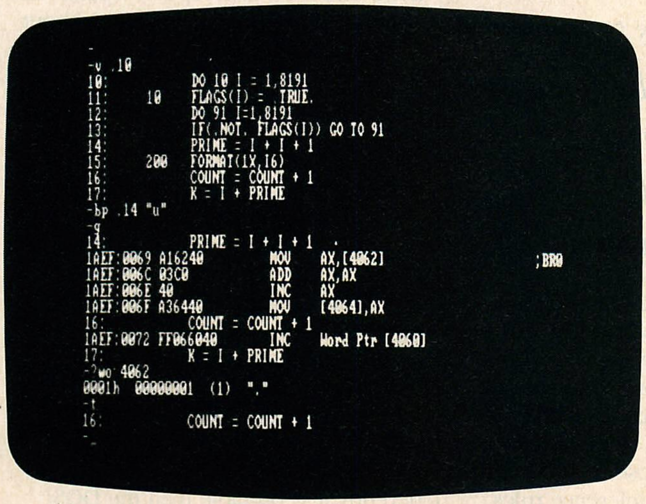
Macro Assembler

The quickest. Bar none.

Our Macro Assembler has long been the most complete package on the market. Now it's also the fastest. Three times faster than before. And faster than anyone else. Period.

Of course, it's still the most powerful assembler on the market. It supports the standard 8086/8087 opcodes. And the new 186/286/287 instruction set. So you can make the most of the new machines.

Debugging is quicker, too. Thanks to our interactive symbolic debugger, SYMDEB. Now you can refer to variables and source code instead of getting lost in hex dumps. And this debugger also works with Microsoft languages like C, FORTRAN and Pascal. So now you can set breakpoints and trace execution—using source code for reference.



Cut your development time dramatically. Microsoft Macro Assembler's Symbolic Debug utility lets you debug your Macro Assembler programs, or debug your Microsoft C, FORTRAN or Pascal programs using your original source code or the resulting disassembly. For example, you can set breakpoints on line numbers and observe the contents of variables or expressions.

SYMDEB is just part of our complete set of utilities. Tools that make programming as fast as it should be. There are the linker and library managers you'd expect. Plus a new version of MAKE, our maintenance utility, with improvements like macro expansions and inference rules.

We've also revised the manuals. Our new Macro Assembler has a lot to offer, so we added more examples. Now our manuals are not only thorough, they're clearer than ever before.

For quick development and assembly, the choice is obvious. Microsoft. There's nobody faster.

Microsoft® Macro Assembler Version 4.0 for MS-DOS®

Macro Assembler
- Fastest macro assembler for MS-DOS computers.
- Supports the 8086/8087/8088 and the 186/286/287.
- Define macros.
- Conditional assembly.
- Optional case sensitivity for symbols.
- 100% upward compatibility from earlier versions of both the Microsoft and IBM® Macro Assemblers.

Interactive Symbolic Debug Utility
- Source level debugger for programs written in Microsoft Macro Assembler, C Compiler, FORTRAN, and Pascal.
- Screen swapping helps debug highly visual applications.
- Set breakpoints on line numbers and symbols.
- Single step to follow program execution.
- Disassemble object code.
- Display and modify values.
- Full I/O redirection.

Program Maintenance Utility
- Rebuilds your applications after your source files have changed.
- Similar to UNIX™ MAKE utility.
- Supports macro definitions and inference rules.

Library Manager
- Create, organize and maintain your object module libraries created with Microsoft languages.
- Set page size from 16 to 32678, to create compact and granular libraries.

Object Code Linker
- Simple overlaying linker combines relocatable object modules created using Microsoft languages into a single program.
- Load Map generation.
- Specify from 1 to 1024 segments.

Cross-Reference Utility
- Creates a cross-reference listing of the definitions and locations of all symbols used in an assembly language program, which makes debugging programs easier.

Microsoft EXE File Compression Utility
- Packs EXE files for smaller size on disk and faster loading at execution time.

Microsoft EXE File Header Utility
- Display and modify EXE file header, allowing you to tune the stack size and initial memory allocation.

For the name of your nearest Microsoft dealer call (800) 426-9400. In Washington State and Alaska, call (206) 882-8088. In Canada, call (416) 673-7638.

Microsoft

The High Performance Software™

one-sixtieth of a second, far faster than the human eye can follow.

The special precautions taken by **cpblk** are not needed when a monochrome adapter and display are being used. Therefore, the driver program, ST, checks for mode 7 and uses a standard block-copy routine that invokes the string copy feature of the 8086/88 processor. A string copy of 4KB is done very quickly. No flicker is visible, and no apparent delay occurs when this approach is used.

Some IBM-compatible microcomputers such as those from Compaq and AT&T have display adapters that have no problems with CPU/raster scan contention. An option on the invocation command line for ST permits a single argument (anything will do) to be given to turn off the use of the special copy feature. The command **st x** tells the program to use a standard block-copy routine instead of the special one. If this option is selected on a system with a standard CGA, visible interference is quite noticeable, especially if the user were to hold down a command key (0-4) to write the same screen buffer repeatedly to the display.

The **getdtype** function (listing 5) uses the ROM BIOS video interrupt to obtain the current video mode. If CI_C86 is defined in the source file or in a command-line option, the C86 **sysint** DOS interface function is used. Otherwise, the **int86** interface function, supported by both Microsoft and Lattice, obtains the needed value.

ST takes the easy way out of a minor dilemma. If a mode of 2, 3, or 7 is returned, the program runs to completion. If any other mode is returned, ST calls the **err_exit** function (listing 6), which displays an error message and returns control to DOS. A more elegant design might save the mode returned by **getdtype**, then switch to an appropriate mode for the duration of the program, and restore the saved mode upon completion. The problem is in knowing which mode to switch to.

Because the Computer Innovations C86 compiler does not recognize the **void** type specifier, a **typedef** is used to alias references to **void** to type **int**. Because no return of either control or a value from an error-exit operation is expected, the conditional compilation route was chosen over using a return type of **int** from the exit function.

To create the ST program using Microsoft C 3.0 or Lattice C 3.0, the programmer should first be sure that the files ST.C, SETPAGE.C, GETDTYPE.C, and ERR_EXIT.C define C_C86 to be 0. Also, the CPBLK.ASM source file must be edited to uncomment the equate for the selected C compiler and comment out the other two equates.

A modest amount of automation is demonstrated here. The MAKE program supplied with the Microsoft Macro Assembler was used to control the development and maintenance of the source files for the screen test program. The make-description file ST is used by MAKE to compile the C and assembly language source files for which no targets exist and for those targets that are out of date with respect to their companion source files. The version of ST shown in listing 7 is specifically for the Microsoft compiler and is invoked by the command **make st**. The compilation and assembly steps produce Microsoft-compatible object files. A macro assembler must be used to assemble **cpblk** because of the conditional assembly language directives within the source file. Both Lattice C and Computer Innovations C86 require that header files be included, so the files must be placed where the assembler can find them. Microsoft MASM 3.01 was used for the assembly step.

The routine described here is inherently less portable than the DOS/BIOS calls, and should be used only for speed and effect.

The MAKE file for ST is written for the MAKE program that does not understand inference rules. Newer versions of MAKE (reportedly, those in MASM 4.0) do permit the use of inference rules and macros, which means the description file can be greatly simplified. Note the **/ml** and **/mx** switches in the **masm** command. These cause the assembler to be case sensitive for internal and external names, respectively. The **/noi** in the link command instructs the linker not to ignore case. For Microsoft C these switch settings are necessary, and for Lattice C they must be deleted from the description file.

The compilers used for this article accept directives to search for header files and object libraries in specified directories. A hard disk is assumed in the presentation of the program preparation steps. The ST file may need editing for configurations different from those used here. The programs were developed and tested on a PC/XT, a PC/AT, and a PC6300, each with a 20MB hard disk and running various color and monochrome display systems.

Compiling, assembling, and linking for Computer Innovations C compiler follows the same pattern as just described above. The programmer must be sure to edit the source files in order to set the necessary definitions and equates. Lattice C uses a start-up object file called C.OBJ, which must be the first file in the object list supplied to the linker. In addition, it requires a DOS.MAC include file that is selected for the memory model being used.

CONSIDERATIONS

Because it takes a tenth of a second to copy data from a screen buffer to display memory, programs should not try to write one character at a time from the keyboard. This would result in a maximum update rate of six characters per second. Even a slow typist would get ahead of such a program.

A better way to handle this situation is to use routines based on the BIOS and DOS interrupts to update the visual display page (they do so without causing interference) and use a separate routine to update the in-memory buffer so that it will reflect what is being displayed. The instant-screen routines should be saved for making large changes to the screen appearance.

Other screen-update routines can be written based on the one described in this article. For example, a routine that copies a single line or a small range of lines or one that copies a small rectangular region from a screen buffer to display memory would be useful in doing selective screen updates with even shorter delays.

The routine described here is inherently less portable than the DOS/BIOS calls, and thus should be used only when necessary for speed and effect. The use of a manifest constant that holds the display memory segment is deliberate. Changing the value for machines that tuck display memory away in some nonstandard place is a simple matter. Program logic that determines what equipment is hosting the program, permitting runtime adjustments to be made, can provide a means of regaining some of the lost portability.

Augie Hansen, formerly on the technical staff of AT&T Information Systems, now owns Omniware, a software development and training company.

LISTING 1: CPBLK.ASM

```
;****** CPBLK *****************************************************
;
;       Copy a screen buffer to display memory without visual "noise"
;       (C) 1982,1986 Omniware
;
;       Usage:
;               cpblk(src_os, src_seg, dest_os, dest_seg);
;               unsigned src_os, src_seg, sest_os, dest_seg;
;
;       Notes:
;       1.  Uses calling conventions Computer Innovations C86, Lattice,
;           and Microsoft C compilers selected by an equate.
;       2.  Provides the copy function solely for the c/g adapter.  Test
;           for the required hardware before calling this function.
;       3.  The segments and offsets given in the calling program
;           determine whether this function does a screen save or a
;           screen update operation.
;*******************************************************************

;-- compiler compatibility (use one of CI_C86, LATTICE, or MICROSOFT) --
;CI_C86         equ     1       ; Computer Innovations, C86 version 2.30A
;LATTICE        equ     1       ; Lattice, version 3.00G
MICROSOFT       equ     1       ; Microsoft, version 3.00

ifdef   CI_C86
        include model.h
        include prologue.h
else
        @ab     equ     4       ; small model only
endif
ifdef   LATTICE
        include dos.mac
endif

a1      equ     @ab
a2      equ     @ab+4

;------ video status information ------
VSTAT   equ     3dah            ; video (CRT) status register
HRTRCE  equ     1               ; horizontal retrace bit mask
VRTRCE  equ     8               ; vertical retrace bit mask
;-- block data (these values result in 8 extra bytes being copied) --
BLKCNT  equ     6               ; no. of buffer blocks to copy
WRDCNT1 equ     240             ; no. of words to copy during ver. retrace
WRDCNT2 equ     94              ; no. of words to copy during raster scan
BYTECNT equ     2 * WRDCNT2     ; no. of bytes to copy during raster scan

ifdef   LATTICE
        pseg
endif

ifdef MICROSOFT
        assume  cs:_TEXT
_TEXT   segment byte    public  'CODE'
        public  _cpblk
_cpblk  proc    near
else
        public  cpblk
cpblk   proc    near
endif
        ;------ save registers and flags ------
        push    bp
        mov     bp,sp
        push    di              ; di and si saved because new C compilers
        push    si              ;   use them for register variables
        push    ds
        push    es
        pushf

        ;------ get the source address ------
        mov     si,a1[bp]
        mov     bx,si
        mov     cl,4            ; shift to extract segment
        shr     bx,cl
        add     bx,a1+2[bp]     ; normalized source segment

        ;------ get the destination address ------
        mov     di,a2[bp]
        mov     dx,di
        mov     cl,4            ; shift to extract segment
        shr     dx,cl
        add     dx,a2+2[bp]     ; normalized destination segment

        mov     ax,0fh
        and     si,ax           ; source pointer
        and     di,ax           ; destination pointer

        cld                     ; set up for auto increment
        mov     ds,bx           ; source segment
        mov     es,dx           ; destination segment
        mov     ah,BLKCNT       ; number of blocks to move

        ;------ COPY A BLOCK -----------------------------------------
        ;
        ; The buffer is copied to the display memory in blocks.  Each block
        ; is copied in two parts.  First, a chunk of words (character and
        ; attribute) is copied during the vertical retrace period and then
        ; individual words are copied during the horizontal retrace periods
        ; of the normal screen update period.  The display is not blanked.
        ;-------------------------------------------------------------

copy_block:
        ;----- copy character/attribute words during vertical retrace -----
        mov     cx,WRDCNT1              ; number of words to copy
        mov     dx,VSTAT                ; c/g adapter status register
wait_vert_refresh:
        in      al,dx                   ; read status
        test    al,VRTRCE               ; test vertical retrace bit
        jnz     wait_vert_refresh       ; loop until in a refresh period

wait_vert_retrace:
        in      al,dx                   ; read status
        test    al,VRTRCE               ; test vertical retrace bit
        jz      wait_vert_retrace       ; loop until retrace starts
        rep     movsw                   ; move a block of char/attr words

        ;------ copy single bytes during horizontal retrace periods ------
        mov     cx,BYTECNT              ; number of bytes to copy
        cmp     cx,0                    ; anything to copy?
        jz      short bypass_horiz      ; no -bypass horiz. period updates
        mov     dx,VSTAT                ; read c/g adapter status register

wait_horiz_refresh:
        in      al,dx
        test    al,HRTRCE               ; test horizontal retrace bit
        jnz     wait_horiz_refresh      ; loop until not in a retrace period
        cli                             ; can't tolerate an interrupt here

wait_horiz_retrace:
        in      al,dx
        test    al,HRTRCE               ; test horizontal retrace bit
        jz      wait_horiz_retrace      ; loop until retrace starts

        movsb                           ; copy a byte
        sti                             ; interrupts OK now

        loop    wait_horiz_refresh

bypass_horiz:
        ;------ see if all rows have been copied ------
        dec     ah                      ; reduce the block count
        cmp     ah,0                    ; done?
        jnz     short copy_block        ; no - do it again

        ;------ clean up and return to caller ------
        popf                            ; yes - restore flags...
        pop     es                      ;   ...and the registers
        pop     ds
        pop     si
        pop     di
        pop     bp
        ret

ifdef MICROSOFT
_cpblk  endp
_TEXT   ends
else
cpblk   endp
endif
```

```
ifdef   LATTICE
        endps
endif

ifdef   CI_C86
        include epilogue.h
endif
        end
```

LISTING 2: ST.C

```
/************************************************************************
 *  st -- screen test using cpblk function
 *
 *  Author: Augie Hansen
 *  Written: 09/10/84
 *
 *  Copies buffer contents into the c/g adapter's display memory
 *  while eliminating "snow" by writing only during retrace periods.
 *  Writes to on undisplayed page (active) and then flips the active
 *  and visual pages to obtain a truely instant update.
 *
 *  --- Revision record ---
 *  02/12/86: Added page-flipping feature for instant screens and
 *      code to determine the display system type in use.
 *  03/20/86: Revised for use with Microsoft C, Version 3.00 and
 *      Lattice C, Version 3.00G
 ************************************************************************/

/* set the following define to 1 for CI C86; 0 otherwise */
#define CI_C86  0

#include <stdio.h>
#include <dos.h>

#define ASCII           0x7F
#define ATTR            0x17
#define CG_SEG          0xB800
#define CGA_C80         3
#define CGA_M80         2
#define MONO            7
#define MONO_SEG        0xB000
#define PAGESIZ         0x1000
#define PG0_OS          0
#define PG1_OS          PG0_OS + PAGESIZ
#define ESC             27

#if CI_C86
#define MOVEIT(a, b, c, d, e)   movblock(a, b, c, d, e);
typedef int void;
#else
#define MOVEIT(a, b, c, d, e)   movedata(b, a, d, c, e);
#endif

int Apage, Vpage;       /* active and visual display pages */

main(argc, argv)
int argc;
char *argv[];
{
    int ca;             /* character/attribute pair */
    int *cap;           /* char/attr pointer */
    int ch;             /* user command character */
    unsigned dseg;      /* destination buffer segment */
    int dtype;          /* display system type */
    int os;             /* page offset in bytes */
    int *sbuf[5];       /* array of screen buffer pointers */
    int sn;             /* screen number */
    int special = 1;    /* use special copy routine */
    unsigned src;       /* source buffer */
    unsigned sseg;      /* source segment */

    /* segment register values */
#if CI_C86
    struct segregs sr;
#else
    struct SREGS sr;
#endif

    void err_exit();    /* error handler */
    char *malloc();     /* memory allocator */
    void movedata();    /* intersegment block copy */
    void segread();     /* get segment reg values */
    void swap_pg();     /* swap display pages */

    static char pgm[] = { "st" };       /* program name */

    /* initialize destination segment */
    if ((dtype = getdtype()) == CGA_C80 || dtype == CGA_M80)
        dseg = CG_SEG;
    else if (dtype == MONO) {
        dseg = MONO_SEG;
        special = 0;
    } else
        err_exit(pgm, "requires 80-column text mode", 1);

    /* process command line */
    if (argc > 2)
        err_exit(pgm, "usage -- st [x]", 2);
    else if (argc == 2)
        special = 0;    /* bypass special block move */

    /* get data segment value */
    segread(&sr);
#if CI_C86
    sseg = sr.sds;
#else
    sseg = sr.ds;
#endif

    /* set up "active" and "visual" display pages */
    Apage = 1;      /* page being written to */
    Vpage = 0;      /* page being viewed */

    /* create the demonstration screen buffers in memory */
    for (sn = 0; sn < 5; ++sn) {
        /* form the fill character/attribute pair */
        switch (sn) {
        case 0: ca = 0x1730; break;     /* '0', wht on blu */
        case 1: ca = 0x0631; break;     /* '1', brn on blk */
        case 2: ca = 0x6E32; break;     /* '2', yel on brn */
        case 3: ca = 0x2033; break;     /* '3', blk on grn */
        case 4: ca = 0x4734; break;     /* '4', wht on red */
        }

        /* allocate a screen buffer */
        if ((sbuf[sn] = (int *) malloc(PAGESIZ)) == (int *) NULL)
                err_exit(pgm, "out of memory", 3);

        /* fill the buffer */
        cap = sbuf[sn];
        while (cap < sbuf[sn] + (PAGESIZ >> 1))
                *cap++ = ca;
    }

    /* display buffers on user's command */
    fprintf(stderr, "\n 0-4 for screen buffers, Esc=exit \n");
    while ((ch = wait_ch()) != ESC) {
        if (ch >= '0' && ch <= '4' ) {
            if (dtype == MONO)
                os = 0;
            else
                os = (Apage == 0) ? PG0_OS : PG1_OS;
            src = (unsigned) sbuf[ch - 0x30];
            if (special)
                cpblk(src, sseg, os, dseg);
            else
                MOVEIT(src, sseg, os, dseg, PAGESIZ);
            if (dtype != MONO)
                swap_pg();
        } else
            fprintf(stderr, " 0-4 for screen buffers, Esc=exit \n");
    }

    /* restore normal visual page and return to DOS */
    setpage(0);
    exit(0);
} /* end main() */
/*
 *  swap_pg -- exchange the global "active" and "visual" page
 *          values and switch to the new visual page
 */
```

You know John Kemeny and Tom Kurtz. They developed the original BASIC. Now they're back with an even better version: a flexible, easy-to-use structured programming language they call True BASIC.™

True BASIC offers a full selection of control structures. External procedures can be compiled into libraries, making True BASIC faster and more powerful than any other programming language.

True BASIC has a complete matrix algebra package and the best graphics ever in a higher level language. And there are optional libraries for things like sorting and searching and 3-D graphics.

True BASIC programs run on any computer which runs True BASIC, good news for users with more than one kind of PC.

You'll love True BASIC. Whether you're programming for your own applications, teaching others, or developing products to go to market, send in the coupon below to receive a free demo disk.

Available for the IBM-PC™ and compatibles, Apple Macintosh,™ and Commodore Amiga.™

To order, talk to your local dealer or call (603) 643-3882 TODAY!

True BASIC™ inc.
39 South Main Street,
Hanover, NH 03755 (603) 643-3882

My computer is: ☐ IBM-PC/compatible
☐ Apple Macintosh
☐ Commodore Amiga

☐ I'D LIKE TO GET MY FREE DEMO DISK.
☐ I'd like more information on the True BASIC language products.
☐ I'm ready to buy. Call me and tell me how.
☐ I'd like information on True BASIC Mathematics Series and other products.

Name____________
Title____________
Company/University____________
Address____________
City, State, ZIP____________
Telephone____________

Suggested retail price $149.90
Educational site licenses available from Addison-Wesley Publishing.

True BASIC Language System is a trademark of True Basic, Inc.
Macintosh is a trademark of Apple Computer Corp.
Amiga is a trademark of Commodore Business Machines.
IBM is a trademark of International Business Machines Corp.

CIRCLE NO. 191 ON READER SERVICE CARD

```
void swap_pg()
{
    int tmp;

    /* exchange the page values */
    tmp = Apage;
    Apage = Vpage;
    Vpage = tmp;
    /* switch to the new visual page */
    setpage(Vpage);
} /* end swap_pg() */
```

LISTING 3: WAIT_CH.C

```
/**********************************************************************
 *  wait_ch -- Get next character from keybd buffer.
 *
 *  If none is ready, the function waits for an input before returning
 *  to the caller. If the input is non-null, it is returned as a 7-bit
 *  ASCII code.  If it is null, the next character is read and returned
 *  as an 8-bit code with the MSB set to indicate an extended scan code.
 *  Responds to Ctrl-Break input.
 *
 **********************************************************************/

#include <stdio.h>

#define ASCII           0x7F
#define HIBIT           0x80
#define KEYIN_CB        8

int wait_ch()
{
        int ch;

        if ((ch = bdos(KEYIN_CB) & ASCII) == '\0')
                ch = bdos(KEYIN_CB) | HIBIT;
        return (ch);
} /* end wait_ch() */
```

LISTING 4: SETPAGE.C

```
/*
 *  setpage -- select "visual" screen page.  Don't believe what you
 *      read in the IBM Tech Ref -- this is definitely not the "active"
 *      display page (where active and visual definitions are those used
 *      in the Microsoft/IBM BASIC SCREEN statement).
 */

/* set following define to 1 for CI C86; 0 otherwise */
#define CI_C86  0

#include <dos.h>

#define VIDEO_IO        0x10
#define SET_PAGE        5

int setpage(pg)
int pg;         /* visual screen page */
{
#if CI_C86
        unsigned char vec = VIDEO_IO;
        struct regval {
                int ax, bx, cx, dx, si, di, ds, es;
        } srv, rrv;

        srv.ax = (SET_PAGE << 8) | pg;
        return (sysint(vec, &srv, &rrv));
#else
        int intno = VIDEO_IO;
        union REGS inregs, outregs;

        inregs.h.ah = SET_PAGE;
        inregs.h.al = pg;
        return (int86(intno, &inregs, &outregs));
#endif
} /* end setpage() */
```

LISTING 5: GETDTYPE.C

```
/*
 *  getdtype -- retrieve the screen mode value
 */
/* set following define to 1 for CI C86; 0 otherwise */
#define CI_C86  0

#include <dos.h>

#define VIDEO_IO        0x10
#define GET_STATE       15
#define LOBYTE          0xFF

int getdtype()
{
#if CI_C86
        unsigned char vec = VIDEO_IO;
        struct regval {
                int ax, bx, cx, dx, si, di, ds, es;
        } srv, rrv;
        srv.ax = GET_STATE << 8;
        sysint(vec, &srv, &rrv);
        return (rrv.ax & LOBYTE);
#else
        int intno = VIDEO_IO;
        union REGS inregs, outregs;

        inregs.h.ah = GET_STATE;
        int86(intno, &inregs, &outregs);
        return (outregs.h.al);
#endif
} /* end getdtype() */
```

LISTING 6: ERR_EXIT.C

```
/*
 *  err_exit -- issue a diagnostic message and terminate
 */
/* set following define to 1 for CI C86; 0 otherwise */
#define CI_C86  0

#include <stdio.h>

#if CI_C86
typedef int void;
#endif
void err_exit(pname, msg, level)
char *pname, *msg;
unsigned int level;
{
        fprintf(stderr, "%s: %s\n", pname, msg);
        exit(level);
} /* end err_exit() */
```

LISTING 7: ST

```
# makefile for ScreenTest (st) program
st.obj:         st.c
        msc st;

cpblk.obj:      cpblk.asm
        masm cpblk /ml /mx;

err_exit.obj:   err_exit.c
        msc err_exit;

getdtype.obj:   getdtype.c
        msc getdtype;

setpage.obj:    setpage.c
        msc setpage;

wait_ch.obj:    wait_ch.c
        msc wait_ch;

st.exe:         st.obj cpblk.obj err_exit.obj getdtype.obj \
        setpage.obj wait_ch.obj
        link st err_exit getdtype setpage wait_ch cpblk /noi, st,,;
```


CIRCLE NO. 137 ON READER SERVICE CARD

IBM®

Equation Solving by Formula/One

An equation-solving program with built-in functions and a textbook-style interface automates mathematical modeling.

VICTOR E. WRIGHT

The computer spreadsheet, widely used in the personal and business worlds, is too limited for most scientific, engineering, and business applications. In some situations, a system or process must be modeled with a set of mathematical relations. Often, the user (or perhaps more appropriately, the modeler) is not interested in creating a computer program to implement the model, nor is it worthwhile for him to transform an existing mathematical model into the methods that could be used in a spreadsheet. What the modeler needs is a tool that recognizes mathematical equations and associated data and that solves the equations.

Equation-solving programs are not new. Classical methods of solving sets of simultaneous equations have been implemented in several languages. The program considered here, Formula/One, from Alloy Computer Products, Inc., is an equation solver patterned after the pioneer in this field, TK!Solver. A sidebar comparing the two follows this article. (Also see "TK!Solver Equates," Victor E. Wright, *PC Tech Journal*, September 1985, p. 137).

Mathematical models are based on *relations*—equations, conditions, constraints, and functions. By contrast, traditional computer algorithms are based on much lower level constructs. Even the statements of a program written in a high-level language, such as FORTRAN, C, or PL/1, are low level in comparison to the relations expressed by the equations these programs solve.

An equation-solving program accepts equations in essentially textbook format, with a few concessions to facilitate the task of parsing. The program "knows" nothing of particle physics, pricing theory, Reynolds numbers, or cantilever beams. What it does know is how to solve equations—*true* equations with an arbitrary number of variables on each side of the equal sign and, in some cases, systems of simultaneous equations. This kind of program is an example of a constraint language;

PHOTOGRAPH • MARC DAVID COHEN

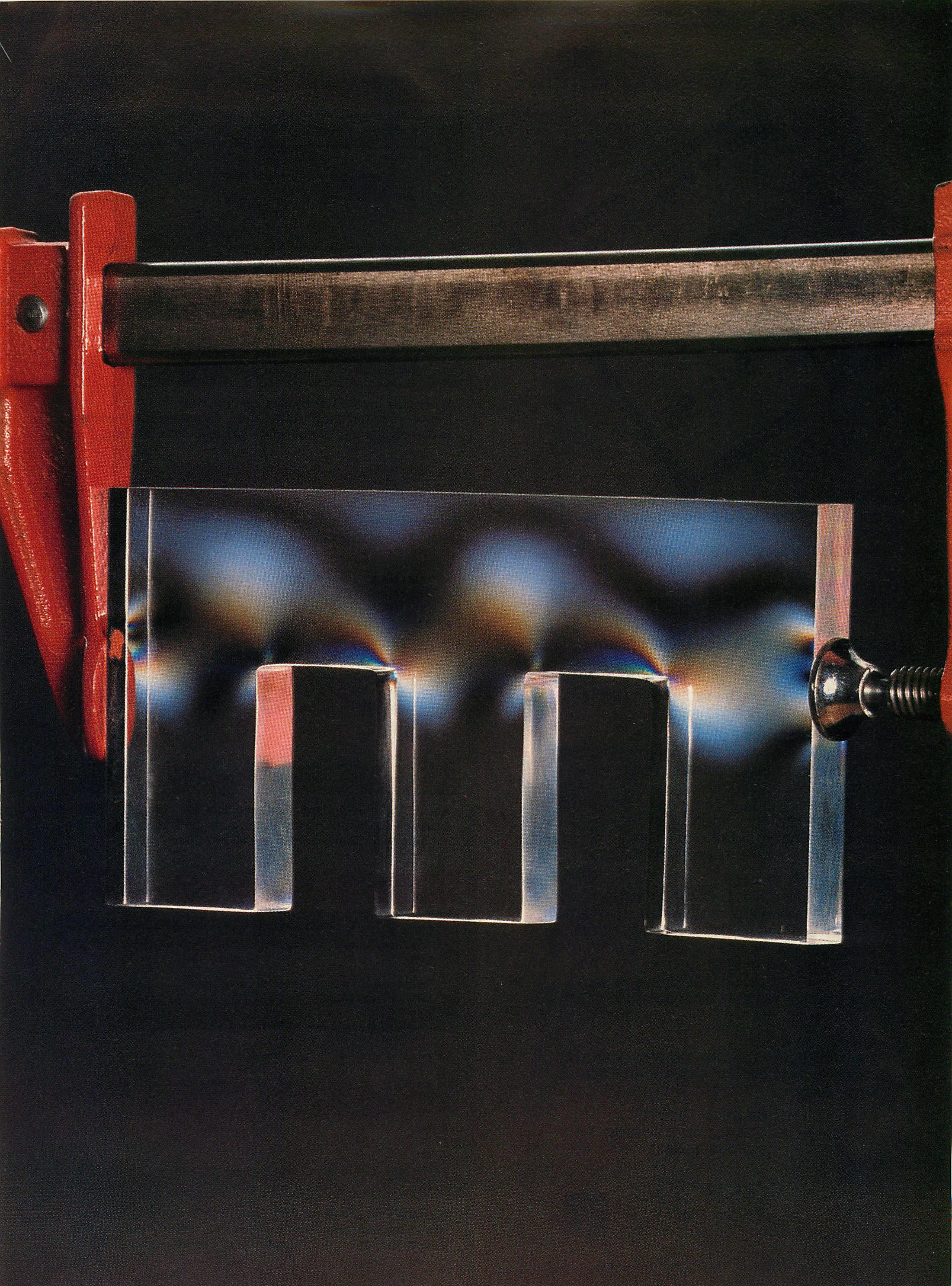
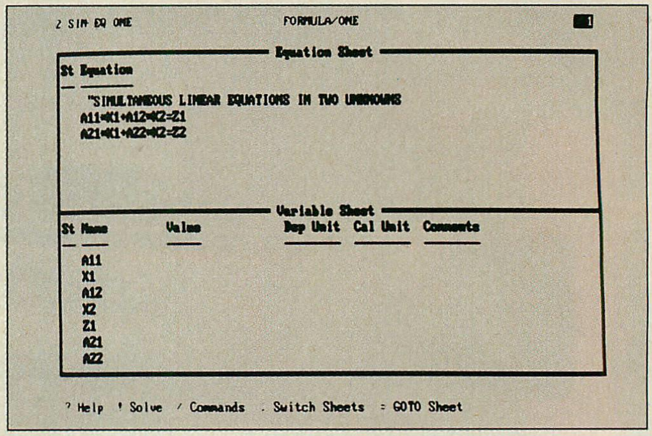


PHOTO 1: *Initial Display*



When first invoked, Formula/One displays a two-sheet format: equation and variable. Here, the modeler has entered two equations and Formula/One has listed the variables.

FIGURE 1: *Specifying Values*

```
============================== Equation Sheet ==========================

St Equation
-- --------
    "SIMULTANEOUS LINEAR EQUATIONS IN TWO UNKNOWNS
   A11*X1+A12*X2=Z1
   A21*X1+A22*X2=Z2
============================== Variable Sheet ==========================

St Name         Value           Dsp Unit  Cal Unit  Comments
-- ----         -----           --------  --------  --------
I  A11          2
   X1
I  A12          1
   X2
I  Z1           13
I  A21          4
I  A22          -2
I  Z2           -2
```

Above, the modeler specifies values for the constants in the equations shown in photo 1. An "I" in the status field of the variable sheet indicates values that have been input.

FIGURE 2: *Automated Solutions*

```
============================== Equation Sheet ==========================

St Equation
-- --------
    "SIMULTANEOUS LINEAR EQUATIONS IN TWO UNKNOWNS
*  A11*X1+A12*X2=Z1
*  A21*X1+A22*X2=Z2
============================== Variable Sheet ==========================

St Name         Value           Dsp Unit  Cal Unit  Comments
-- ----         -----           --------  --------  --------
I  A11          2
O  X1           3
I  A12          1
O  X2           7
I  Z1           13
I  A21          4
I  A22          -2
I  Z2           -2
```

Formula/One solves for two unknowns in figure 1. The asterisks in the equation sheet indicate that both equations were solved; the Os on the variable sheet denote output values.

it relies upon the relations that describe the problem and determine the solution. The rules of an such a program are the rules of a universally understood language: mathematics.

FAMILIAR INTERFACE

The Formula/One interface allows a model to be defined in terms of mathematical equations in the same sense that equations are presented in references and textbooks. The modeler enters true equations; the program accepts the known values and assigns values to unknown variables such that the conditions specified by the equations are satisfied. A built-in solver of unspecified methodology performs the work.

This equation solver uses the notebook and sheets paradigm: that is, a Formula/One model is defined by a number of relations, each of which is summarized in a specific screen display that is like a sheet in a scientist's notebook. Formula/One is interactive—a what-if tool. Once a model is defined, it can be solved with many sets of input. A model can be developed incrementally and solved at any stage.

When the program signs on, it asks what type of monitor is being used. Once this is resolved, the program displays two windows into the model—the equation sheet and the variable sheet. These two sheets are required for even the most minimal modeling. Photo 1 shows a typical configuration with two equations entered. The equation sheet (at top) contains the model's equations; variable names are listed automatically on the variable sheet.

The equation sheet has two fields: status and equation. The window can display either seven rows when the screen is split between the two sheets, or seventeen rows when the screen is toggled to a single-sheet display. The remaining rows are devoted to borders and field headings. A reverse video cursor appears in the first row of the equation sheet in the initial display, indicating that the equation will be entered there. The cursor can be moved to any row with the cursor arrow keys.

When a key is pressed, with the cursor in the equation field of the equation sheet, a reverse video bar appears at the bottom of the screen. As the current equation is typed in, it appears in the bar, not in the sheet itself. When the equation is entered, by pressing the Return key or an up or down arrow key, it appears in the current row, and the bar at the bottom disappears. Formula/One checks the equations for legal syntax as they are entered.

The variable sheet is comprised of six fields: status, name, value, display unit, calculation unit, and comments. Like the equation sheet, the variable sheet displays seven rows in a dual-sheet display and seventeen rows as a single sheet. As equations are entered in the equation sheet, the variable names appear in the variable sheet name field. Variable names are limited to 16 characters, 12 of which can be displayed in the name field.

The status field displays the type and use of each variable: I(nput), O(utput), G(uess), L(ist), I(nput) L(ist), O(utput) L(ist), or S(tring). This range of options enables Formula/One to use a single column to indicate input and output values.

The number of required input values depends upon the number of independent equations, and the model can have as many unknown variables as it has independent simultaneous equations. The entering of variable values is similar to the entering of equations. The cursor must be positioned in the row of the current variable. (In the case of a new model, the cursor first must be moved from the equation sheet to the variable sheet with the switch sheet command—the ; key). As input values are typed in, they first appear in the reverse-video bar at the bottom of the screen, then in the current row when the data are entered. As the entry appears in the proper row, the program displays an I indicator in the status field. Figure 1, which is a continuation of the Formula/One session shown in photo 1, demonstrates input.

Equations, variable names, and input values are the essential elements of a model. Once they are entered, the model can be solved using the solve command (the ! key) or by pressing F9. As the program solves the model, it indicates the progress of the solution process on the bottom line of the screen. When the solution is complete, the output values appear in the value field of the variable sheet, with O indicators in the status field of each output variable, and an asterisk (*) appears in the status field of the equation sheet for each equation that was satisfied. In figure 2, Formula/One has solved the equations from figure 1.

UNITS

Units are a key element of any model. A relation can be described in an abstract sense; but without units, it cannot be related to real-world phenomena. Moreover, if units are not supplied, a model cannot be considered *truly* solved.

Formula/One assumes that a model is entered with a consistent set of units. Even if a model is developed without units, it can be solved and output values will be displayed: if the equations that form the model are consistent (incorporating conversion factors and constants as required), then the model will produce correct answers, whether or not units are assigned to variables.

The unit sheet holds the definition of unit conversion factors. A unit conversion is defined by information in four fields: from unit, to unit, multiply by, and add offset. These fields permit a linear transformation between any two units of measurement. As many as 128 conversion factors can be defined.

Formula/One does not associate the operations applied to variables with corresponding operations to units, so it cannot distinguish among suitable units. For example, if an equation for area, such as $A = l * w$, has the unit feet assigned to l (length) and w (width) and the unit inches^2 assigned to A, the equation will be solved without comment, just as it would if A had been assigned the more logical unit of feet^2.

In fact, Formula/One does not monitor the status of unit assignments on a realtime basis. The variable sheet includes two fields for units—calculation and display—both of which accept entries of arbitrary unit labels at any time, whether or not they are currently defined in the unit sheet. The program apparently checks the unit sheet for conversion factors only during the solution process. If a display unit label is changed, the corresponding value does not change until the model is solved again. When the model is solved, conversions are made only as defined in the unit sheet. If a conversion has not been defined for a particular display/calculation unit pair, the value is displayed in calculation units. A conversion must be defined for each pair—the program does not chain conversion factors.

The equation, variable, and unit sheets provide enough horsepower to solve many engineering problems. In figure 3, Formula/One solves a stress analysis problem (that is illustrated and solved conventionally in the sidebar "A Sample Problem" on page 123).

LISTS

A variable can be assigned a L(ist) status when, for example, a single model must be solved for several sets of input data, or when a single model requires a numerical solution in which each step of the solution requires access to the data of previous steps (see figure 4).

Lists are created from the variable sheet, by entering L or IL in the status field; they can contain numerical or string values, but not both. The modeler reaches the list sheets that are created by using the dive command (>) from the variable sheet. String values may be held only in lists that have an S in the status field; if any other list indicator, or no indicator, appears, string values cannot be entered.

Once a list has been created, it can be filled with input values in two different ways. In the first method, the modeler enters individual values, one row at a time, in any arbitrary sequence. The second method uses the fill command (/F). Values entered using fill are spaced linearly by the program between the first and last entries in the list, both of which must be entered directly. For example, if the modeler enters 1, then four blank lines, then 6 as a list, and presses /F, Formula/One will fill in the list with 2 3 4 5, each on its own line between those holding 1 and 6.

Each list sheet has a number-of-list-elements field that must be set before /F is entered, because the command fills the entire list, not an arbitrary subrange. Individual list sizes can be set up in this way, or a default list size, up to 999 elements, can be set for all lists in a

FIGURE 3: *A Sample Solution*

```
============================== Equation Sheet ==========================

St Equation
-- --------
    "Beer & Johnston, STATICS, page 131
    "Weights=forces
   Fload=Load*g
   Fcrane=Mcrane*g
    "Moments about B
   -1.5*Ax-6*Fload-2*Fcrane=0
    "Moments about A
   1.5*Bx-6*Fload-2*Fcrane=0
    "Sum of forces in X direction
   Ax+Bx=0
    "Sum of forces in Y direction
   Ay+By-Fload-Fcrane=0
============================== Variable Sheet ==========================

St Name        Value         Dsp Unit  Cal Unit  Comments
-- ----        -----         --------  --------  --------
I  Load        2400          kg        kg
I  g           9.81          m/s^2     m/s^2
O  Fload       23544         Newton    Newton
O  Fcrane      9810          Newton    Newton
I  Mcrane      1000          kg        kg
O  Ax          -107.256      kN        Newton    pinned connection
O  Bx          107.256       kN        Newton    rocker connection
O  Ay          33.354        kN        Newton    pinned connection
I  By          0             kN        Newton    rocker connection = 0

=============================== Unit Sheet ============================

From Unit   To Unit    Multiply By     Add Offset
--------    -------    -----------     ---------
kN          Newton     1000
```

In figure 3, Formula/One solves a stress analysis problem that is worked out conventionally in the sidebar "A Sample Problem" (p. 123). In figure 4, the program processes a *list* of values. If an object is dropped at time $t = 0$, the distance S it has traveled is given by $S = \frac{1}{2}at^2$, where a is the acceleration due to gravity. Formula/One finds S at $t = 1$ second, $t = 2$ seconds, and so on, with a single solve command.

FIGURE 4: *Using Formula/One Lists*

```
============================== Equation Sheet ==========================

St Equation
-- --------
*  S=1/2*a*t^2
============================== Variable Sheet ==========================

St Name        Value         Dsp Unit  Cal Unit  Comments
-- ----        -----         --------  --------  --------
OL S
I  a           9.8
IL t

================================ List Sheet ============================

Number of List Elements:      10
t
-
1
2
3
4
5
6
7
8
9
10
================================ List Sheet ============================

Number of List Elements:      10
S
-
4.9
19.6
44.1
78.4
122.5
176.4
240.1
313.6
396.9
490
```

model (this is effected from the set-up sheet, which is discussed below).

Any list not indicated specifically as an I(nput) L(ist) is assumed to be an O(utput) L(ist); the program changes the L to OL when the model is solved. Regardless of input or output, as long as list status is indicated, input values cannot be entered in the value field.

Formula/One will not solve a model in which all of the variables are assigned L, the general list status. At least one list must be assigned IL status. The program has no list command; when a model contains input lists, the list solver is invoked automatically when the solve command is entered.

Lists can be used as arguments of certain functions in a model that is to be solved only once—a model, that is, without input and output lists. In that case, list variables cannot appear in the equations as variables. Instead, the list name must be entered in the name field of the variable sheet directly, and then the list status is set.

SOLUTIONS

Formula/One provides a single solve command. The modeler enters this command and the program determines the solution method from various aspects of the model. Its primary method is the algebraic solver (analogous to TK!Solver's direct solver).

This solver proceeds through the model, beginning with the first equation, searching for equations with only one unknown variable to solve. It assumes the substitution of known values from earlier equations. Certain conditions must be met for the algebraic solver to solve a system of equations:

- First, an unknown variable must appear only once in the equation. For maximum efficiency, this rule should be followed throughout a model. Many equations that contain variables in more than one location in textbook form can be rewritten so that the variables appear only once. For example, $y = x * x$ cannot be solved by the algebraic solver, but $y = x^2$ can.
- Second, as was indicated above, each equation can have no more than one unknown variable.
- Third, the unknown variable cannot be the argument of a function for which there is not a unique inverse. For example, the built-in function **sqrt(x)** has a unique inverse, therefore $y = x^2$ can be solved directly whether x or y is the known variable. However, $y = sin(x)$ can be solved directly only when x is known, because the sine function is periodic.

If the algebraic solver is unable to solve an equation or a system of equations (or when the conditions required for the algebraic solver have not been met), the program invokes an iterative solver. Like virtually all iterative methods, Formula/One's method makes a guess at the value of the unknown variable, evaluates both sides of the equation, and compares the two sides. If the two values are equal or differ by less than a predetermined amount, the guess is presented as the correct an-

swer: the output value. If the two sides differ by too great an amount, a new guess is generated, based on the previous guess and according to an unspecified rule, and the process is repeated.

Formula/One uses a single criterion for evaluating the correctness of an iteration—a *comparison tolerance*. This value, .000001 by default, is stored in the set-up sheet. The length of the iterative process is controlled by both the comparison tolerance value and a *maximum iteration count*, also stored in the set-up sheet. The default for this count is 10 but it can be changed to any value from 1 to 999.

With no separate list command, Formula/One determines the need for list processing based on the status fields of the variable sheet and repeats the solution process for each element of the shortest list—not including lists that appear only as arguments of list functions. That is, the list solver repeats the solution process as long as it has input values and places to store them. If all the equations in a model do not reference all the list variables, the number of solutions may not be the same for each equation. If the first equation can be solved directly from the first set of input values, it will be solved repeatedly for each element in the shortest list before the second equation is solved.

Formula/One does not complain about inconsistent or over-constrained models. When an equation is first defined, its status indicator is blank. After the model is solved, each equation will have one of two indicators displayed in its status field: a * meaning solved or a ? meaning unsolved. No indication is given if a model was overdefined (which is usually the case if input values are assigned so that the number of unknowns is less than the number of independent simultaneous equations). In some cases, the program will display a status indicator of Z, with an accompanying message, "Zero determinant, equations may be inconsistent."

Consequently, care must be taken in assigning input values, especially in models that require iteration for a correct solution. If too many input values are assigned in such a model, the program may announce a solution solely with the use of the algebraic solver on selected equations, ignoring the remainder. Fortunately, the program flags the unused equations with the ? indicator.

The solve command is supplemented by the solve new command (F4), which is mentioned in the help screen on solving and in the quick reference card, but not in the manual. This

command solves any equations left unsolved by the first command, assuming that action has been taken to correct the condition that prevented a complete solution. If all equations are unsolved, the effect of the solve new command is identical to that of the solve command.

THE RESULTANT DISPLAY

For models that do not use lists (and following a successful solution), Formula/One displays the values of all variables in their respective value fields. The status field indicates which values were input by the modeler and which were output by the program. Results are displayed in display units if the appropriate conversion factors were defined in the units sheet and in calculation units otherwise.

A second method of presenting results is the display sheet, which lets the user define a custom input/output screen—the generated display. The display sheet has four fields—header 1, name, format, and header 2. It can have 92 lines, with each page displaying 23 (line 24 holds error messages; line 25 lists function key assignments).

The generated display can make the user interface to a model even friendlier than the standard equation/variable sheet display. Long field labels and instructions can be displayed to facilitate the use of the model by persons who have not learned to use the program. The cursor can be positioned only in data fields. Input values can be supplied, the model solved, and tables and plots displayed from the generated display. The modeler reaches the generated display via the dive command from the display sheet.

For models that use lists, output lists can be examined singly by diving to the specific list sheet from the variable sheet. The preferred method to examine the data of a list-based model, however, is to define tables or plots to display the list data. Tables and plots are defined in separate sheets.

The table sheet permits specification of the display device (screen or printer), horizontal or vertical layout, header underline, a sort variable and sort order, the title, and the lists that will comprise the table. A table can include as many as 10 lists. The format of each one can be specified individually, and a one- or two-line header can be displayed above each list. To display the current table—the one defined on the table sheet—the modeler uses the dive command (or F8).

The plot sheet defines graphs or plots. Similar to the table sheet, the plot

sheet permits specification of the display device (screen or printer), as well as the status of the display of scales along the x and y axes. One field specifies the list variable associated with the x axis, and up to 10 list variables can be specified for display on the y axis.

Formula/One plots are produced with characters, not as graphics, regardless of the graphics capabilities of the display device. This is true for both screen and printer displays. Plots can be produced by pressing dive (>) from the plot sheet or F10 from any other sheet in the model.

NAVIGATING

Most models require the use of several sheets, any two of which can be displayed on the screen at any given time. Several commands are devoted to moving around within the model.

As discussed above, the session begins with the display of the equation and variable sheets, with the cursor positioned in the first row of the equation sheet. The cursor arrow keys, along with Home, End, PgUp, and PgDn, are used to move the cursor around the current sheet. The switch sheet command moves the cursor between sheets in a two-sheet display. With these keys, the cursor can be positioned anywhere in the current virtual display, consisting of the entire equation sheet and the rows of the variable sheet that contain entries, plus one blank line immediately below the last entry.

The window command (/W) is used to toggle the display between a single-sheet and a two-sheet format. In changing from a two-sheet display to a single sheet, the sheet in which the cursor is located when the command is executed will remain on the screen. In moving from a single sheet to a two-sheet display, the sheet previously removed is returned.

To change the sheet completely from any of the main sheets—equation, variable, display, table, plot, curve, regression analysis, unit, and set-up—the goto command (=) is used. Executing this command (except during the entry of an equation) displays a list of sheet names on the bottom line of the screen. The desired sheet is selected by typing the first letter of its name. Goto is not available directly from list sheets or from plot, table, or generated displays; from any of these sheet, the return key (<) or Esc must be pressed first.

Formula/One includes a context-sensitive help facility. Help is invoked by pressing the ? key, but only when the program is waiting for a command—help is not available during the entry of an equation or value. The help file is hierarchical in organization and quite responsive. The modeler chooses a page by locating the cursor in a specific sheet and field. Each page has paragraphs of text and a menu of additional, related topics. Common to all help pages is a line of instructions for proceeding to the next page (Return), returning to the previous page (^), and leaving the help function (Esc).

FUNCTIONS

Formula/One includes an extensive complement of math, conditional, list, and special functions. The standard math functions are provided—trigonometric, inverse trigonometric, logarithmic, and exponential—and 12 different forms of the IF statement comprise the conditional functions. Table 1 describes the list and special functions.

Functions, along with the appropriate arguments, can be included in expressions wherever a simple variable can be used. Functions generally share the syntax, $Y = f(X)$, where X is a variable, a constant, or an expression, which may contain nested functions.

Formula/One's math functions are straightforward, as are most of its special functions. Several list functions re-

TABLE 1: *Special Functions*

LIST FUNCTIONS	
COUNT(LIST)	Returns the number of elements in list
CORR(LIST1,LIST2)	Returns the correlation coefficient of two lists
DPOLY(X,LIST)	Returns the value of the derivative of the polynomial whose coefficients are stored in list, evaluated at *x*
DOT(LIST1,LIST2)	Returns the dot product of the vectors stored in two lists
IPOLY(X,LIST)	Returns the value of the integral of the polynomial whose coefficients are stored in list, evaluated at *x*
MAX(LIST)	Returns the largest value stored in list
MIN(LIST)	Returns the smallest value stored in list
MVAVG(LIST,X,Y)	Returns the average of *x* elements of list, beginning with element *y*
NPV(R,LIST)	Returns the net present value of list at rate *r*
POLY(X,LIST)	Returns value of polynomial whose coefficients are stored in list, with variable = *x*
STDEV(LIST)	Returns the standard deviation of list
SUM(LIST)	Returns the sum of the elements of list
SUMSQ(LIST)	Returns the sum of the squares of the elements of list
USE(X,LIST)	Returns the value of the *x*th element of list
VAR(LIST)	Returns the variance of list
MISCELLANEOUS FUNCTIONS	
ABS(X)	Returns the absolute value of *x*
ATAN2(X,Y)	Returns the arc tangent of *x/y*
CASHF(X,Y,Z)	Returns the cash flow per term required to produce the net present value *x*, at interest rate *y*, for term *z*
ELT()	Returns current element number during list solve
FACT(X)	Returns *x*!, factorial of *x*
INT(X)	Returns the integer portion of real number *x*
INTEG(FX,X,A,B,N)	Returns the value of the function *fx*, integrated with respect to *x*, over the interval *a* to *b*, using Simpson's Rule—*n* is an optional number of subintervals, which defaults to 10
MOD(X,Y)	Returns the remainder produced by *x/y*
SIGN(X)	Returns 1 if $x > 0$, 0 if $x = 0$, and -1 if $x < 0$

Formula/One is notable for its many specialized functions: mathematical functions for polynomials and statistical functions for series. This program also provides the standard math functions, along with their inverses.

semble those of advanced spreadsheets. The polynomial and element-related functions probably will be new to spreadsheet users moving up to equation solvers. The **elt( )** function is a pointer that provides a value equal to the position of the list solver, as it steps through the lists. This pointer can be used as the base of a calculation to access elements of a list other than the current element. The **use(...)** function actually retrieves the value of a specific element of a list. Used in conjunction with the **elt( )** function, **use(...)** is available for various numerical methods, such as difference functions, numerical differentiation, and integration.

The polynomial functions provide a way of working with integrals and derivatives (assuming the program's built-in method is satisfactory). The statistical functions simplify the construction of financial and statistical models.

The conditional functions simplify the construction of models that must accommodate piecewise functions. Systems that must be described by different equations in different regimes require models composed of several equations using the same variables. Conditional functions can provide logical multipliers to select the proper equations, based on the value of one or more parameters.

Although Formula/One includes a powerful assortment of list functions, all of them are limited by the manner in which the program handles the assignment of variable values. As noted above, a variable cannot be assigned a string value unless the its status is set to S; this makes it a list variable. As a result, list functions accept quoted lists only. A variable that holds the name of a list is not a valid parameter; neither is a function that returns a list.

Noticeably absent is a facility for defining user functions. (TK!Solver provides a mechanism to create a user function by associating two lists, one as the domain, the other as the range. Values can be mapped in three ways.) Without such a facility, Formula/One cannot accommodate empirical functions (such as step functions) exactly, although the curve-fitting facilities described below offer some capability.

CURVES AND OTHER SHEETS

Formula/One includes built-in curve-fitting and regression-analysis facilities, with individual sheets devoted to each. The curve sheet has five fields that define the curve-fitting parameters. The first two fields—dependent variable and independent variable—are input fields for the names of list variables. Both lists contain values. The third field indicates the name of a list in which the program will store the coefficients of a polynomial that will describe the relation between the two data lists. The fourth field indicates the degree of the polynomial to be fit to the data. The last field, the correlation coefficient, is an output field that displays a measure of the "goodness of fit."

A curve-fitting session begins with the variable sheet. The first step is to enter the names of the desired lists in the name field. Then, each variable is assigned L(ist) status. Next, the size of each list is set to accommodate the required list of values. Finally, the independent and dependent variable lists are filled with values.

The curve sheet is then accessed with the =C command, and the list variable names are entered, along with the degree of the desired polynomial. Entering the solve command (or pressing F9) causes the program to generate the coefficient list and the correlation coefficient. By starting with a low degree, and increasing it, the data can be fit with a curve of the lowest degree that produces an acceptable fit. Once the curve has been fit, the modeler has a polynomial that can be used as an approximation of an empirical relation.

Whereas the curve sheet provides the modeler with a method of fitting a polynomial of degree 9 or less to a dependent variable as a function of a single independent variable, the regression-analysis sheet provides a method of fitting a linear, or first-order function, of several independent variables to a dependent variable. The regression-analysis sheet is similar to the curve sheet, but instead of a single independent variable field, it has 10 rows of fields. Each row has a field for the name of an independent list variable and for a corresponding coefficient. The sheet also contains an output field labeled intercept, which displays the constant term of the resulting polynomial.

The regression-analysis sheet can produce a polynomial that describes the relationship among several lists, and the significance of each independent variable in the relation. The resulting polynomial can be used to predict the value of the dependent variable, or the most significant variables can be used in a simpler polynomial. This facility also accommodates empirical relations.

The set-up sheet contains some previously discussed fields: maximum iteration count, maximum list length, comparison tolerance. These fields determine the global default list size, as well as the parameters that control iteration. Another field, intermediate redisplay, controls whether the results of intermediate calculations are displayed during a solution.

This sheet also has a comment field, for model identification (ID), which is displayed when the disk directory is called from within Formula/One. The remaining fields control communication with the output device. For the PRN list device, the printer width and page length can be set. If the output is to be sent to a disk file for printing at a later time, via the print command (/P) for example, the file name is entered in the printer or file-name field.

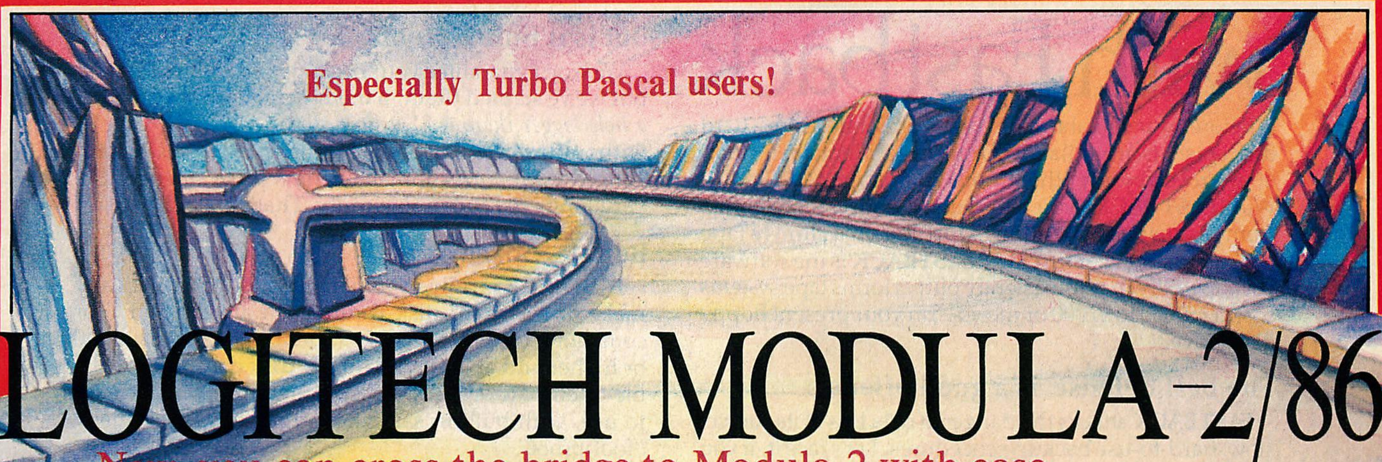
The ability to send printed output to a disk file permits the data from table and plot sheets to be incorporated into word processor data files, and to be edited. The contents of any sheet can be printed using /P, or sent to a disk file. Thus, an entire model can be incorporated into a report, via a word processor data file, or sent to a database management program.

MOVING AND STORAGE

Formula/One models are contained completely in RAM, but they can be saved to disk at any time with the save command (/S); the program requests a name for the storage file. A model can be stored in stages as it is developed.

Models are recalled from disk using the load command (/L), which copies the disk file into memory with no modifications. The load command also will prompt for the name of the file to be loaded. Both the save and load commands will list the model files on the current disk or a specified disk, with ID field data, if available, displayed adjacent to the file name.

This is the Modula-2 compiler *everybody's* been waiting for...



Especially Turbo Pascal users!

LOGITECH MODULA-2/86

Now, you can cross the bridge to Modula-2 with ease.

This is Modula-2 at its *absolute* best. It's a fully integrated development environment that takes into account what you need as a programmer. Without leaving the Editor, you can call the compiler, linker and utilities.

With Logitech's Modula-2, you'll have the ability to edit several files at once, comparing, window to window, various code modules. You can even move from window to window compiling, linking, debugging and running.

The compiler has the kind of power and room to breathe that you really need in today's complex applications. It is as *easy* to use as Turbo Pascal, without your programs being limited to 64K of code.

At your command will be the libraries of modules that make Modula-2 a programmer's dream. It has essentially the same structure as Pascal with the major addition of a library organization of code modules that allow you to put together programs on a solid, block-by-block, foundation of proven code.

Whether you're working with a module of your own making, or one of the many in our library, you'll find the system by which each module is identified, described and stored an organizational masterpiece. And that's at the heart of Modula-2.

Underneath the sophisticated system is a Modula-2 compiler that is the result of years of development and proven use in industry. We run on the Vax*, and we run on the IBM PC. And the code is portable–from one to the other.

Best of all . . . you can have it right now!

Logitech Modula-2/86 Complete with Editor, Run Time System, Linker, Cursor-positioning debugger, 8087 Software Emulation, BCD module, Logitech's extended library, Utility to generate standard .EXE files, Turbo Pascal (and standard Pascal, too) to Modula-2 translator *(included without charge until 8/1/86)*, and much, much more! **$89**

Logitech Modula-2/86 with 8087 support Even if you haven't yet gotten an 8087 co-processor, you can still use this version. **$129**

Logitech Modula-2/86 Plus For machines with 512K or more. Takes advantage of the larger memory to increase compilation speed by 50%! Supports 80186 and 80286 as well as 8086 and 8088. Includes 8087 and 80287 support, too. **$189**

Window Package Now you can build true windowing into your Modula-2/86 code with ease, too. Very powerful and very full, yet only 15K in size. Features virtual screens, color support, overlapping windows and a variety of borders. **$49**

Run Time Debugger (source level) Much more powerful than just a symbolic RTD. Display source code, data, procedure call chain and raw memory. Set break points, assign values to variables, pinpoint and identify bugs in your source. The ultimate professional's tool! **$69**

Utilities Package Features a post-mortem debugger for static debugging. If a program you've written crashes at run time, the situation is frozen, and you can pinpoint, in source, the cause of the error and the data at that moment. Also includes a disassembler, a cross reference utility and a "version" utility that allows conditional compilation. **$49**

Make Utility Automatically selects modules affected by code changes for quick and minimal re-compilation and relinking. Even figures out dependencies for you. **$29**

Library Sources Source code for our major library modules is now available–for customization or exemplification. **$99**

ROM Package If you need to produce rommable code, call our 800 number for further information on this package.

To place an order call our special toll free number

800-231-7717

in California

800-552-8885

Special offer until 8/1/86! **$89**
includes
Free! $49.95 value Turbo Pascal translator!
Now, you can take your library with you!

Yes, I'd like to take the next logical step in programming.

Please send my copy of Logitech Modula-2/86 to the following address:

☐ VISA ☐ MasterCard ☐ Check Enclosed

Card Number ____________ Expiration Date ____________

Signature ____________

Name ____________

Address ____________

City ____________

State ____ Zip ____ Phone (____) ____

Here's the configuration I'd like:

☐ Logitech Modula-2/86 $89
☐ Logitech Modula-2/86 $129 with 8087 support
☐ Logitech Modula-2/86 Plus $189

Please add $6.50 for shipping and handling.

And include the indicated items:

☐ Window Package $49
☐ Run Time Debugger $69 (source level)
☐ Utilities Package $49
☐ Make Utility $29
☐ Library Sources $99

Total enclosed $ ____________
(California residents, please add applicable sales tax)

LOGITECH

LOGITECH, Inc.
805 Veterans Boulevard
Redwood City, California 94063
Telephone (415) 365-9852

For European pricing, please contact:

LOGITECH SA
Box 32, CH-1143 Apples, Switzerland
Telephone 41 (21) 774545

Please call our 800 line for: ☐ Information on our *VAX version ☐ Site License and University Discounts ☐ Dealer and Distributor information

CIRCLE NO. 229 ON READER SERVICE CARD

*Turbo Pascal is a registered trademark of Borland International

Data Defense!

with Fastback*

Before your valuable data comes under fire from a hard disk crash, is annihilated due to an accidental reformat, or destroyed by any other frontal attacks, get **FASTBACK.** With **FASTBACK**, you can defend your valuable data from loss *without* spending a fortune on a clumsy tape backup system. For just $179, **FASTBACK** is the software backup utility that consistently outperforms tape systems costing $1000 to $2000 or more — on your present floppy disk drive.

The best defense is a good offense.

FASTBACK attacks the danger of data loss at its source: slow, hard-to-use backup procedures. Let's face it, if you have to spend *hours* to back up your full hard disk (as you do with the DOS "BACKUP" utility), you're not going to do it very often. But with **FASTBACK** you can back up a full *10 megabytes in just 8 minutes flat (4 minutes on the IBM PC-AT)!* So you can back up every bit of data every day, without even missing a coffee break.

But is it bullet-proof?

We won't guarantee your backups will be bullet-proof, but we've stapled them, nailed them, and *even punched a hole bigger than a .22 calibre slug right through one* — without losing a single byte of data! **FASTBACK's** advanced error-correction algorithms push safety to the very limits of human ingenuity. That's Data Defense!

Be safe, not sorry.

FASTBACK is fully *self-contained,* and works with any make or size hard disk running on any IBM™ PC or compatible (requires PC-DOS™ or MS-DOS™ version 2.0 or higher and at least one floppy disk drive). It is fully file-oriented (not a "disk image"), menu-driven and easy to use, with built-in safeguards that make it virtually "fool-proof".

So get the safe solution, and get it fast. Get **FASTBACK** today.

5 Fifth Generation SYSTEMS

909 Electric Ave Suite 202, Seal Beach, CA, 90740; 7942 Picardy Ave Baton Rouge, LA, 70809; Tel : 1-800-225-2775 or (213) 493-4483

Marketing and Sales by CSSL, Inc.

*This product is in no way associated with or has origination in Fisher Scientific or Allied Corporation. IBM-PC, IBM-XT, IBM-AT and PC-DOS are registered trademarks of International Business Machines Corporation. MS-DOS is a registered trademark of Microsoft, Inc.

CIRCLE NO. 108 ON READER SERVICE CARD

Models can be saved in their entirety or in modules. The options of the save command are: all sheets, the equation sheet, the variable sheet and associated lists, and the unit sheet. These options permit the accumulation of libraries of equations, variables, variable lists, and lists of unit conversion factors. Thus, models can be assembled from a library of modules, instead of being constructed from scratch every time. Loading a model does not cause the current model to be overwritten.

FINISHING TOUCHES

The documentation comes in an IBM-style binder. Most of the material is a tutorial that takes the new user through the entire program using simple models. The distribution disk contains a similar tutorial program. The latter portion of the manual is appendices that contain the essential reference material—function syntax, equation syntax, and so on. Once the modeler learns the commands, the appendices become the most important part of the manual.

The Formula/One package also includes a program on disk called ONE23. This file-conversion utility translates Formula/One data files into a format that can be read by other programs, including spreadsheets and database management systems. The translation is accomplished using comma-delimited data files, which also facilitates the production of user programs to access the model files in BASIC, for example. ONE23 also will transfer data directly to and from Lotus 1-2-3 .WKS files, without having to use intermediate data files. ONE23 can be used in either interactive or batch mode.

Formula/One is written expressly for the IBM PC and compatibles. Its hardware requirements are one floppy-disk drive, 192KB of RAM, and an 80-column display, color or monochrome. The program is copy protected (but it can be installed on a hard disk). A hard disk will load, save, and retrieve models faster, but because a model is contained completely in RAM, the hard disk offers no speed advantage to creating and solving. Formula/One was reviewed on a PC, a PC/AT, and the AT-compatible Heath/Zenith H/Z-200. All three exceeded the minimal hardware requirements; no problems were encountered. Two distribution disks are furnished, and each disk can be installed twice; the program also can be uninstalled.

The model designer who thinks in terms of mathematical constraints and equations will like the Formula/One approach. This program offers a most straightforward transformation of textbook-style mathematical models into computer-automated form.

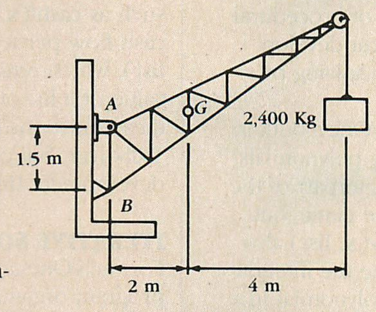
Formula/One: $395
Alloy Computer Products, Inc.
100 Pennsylvania Avenue
Framingham, MA 01701
617/875-6100

CIRCLE 347 ON READER SERVICE CARD

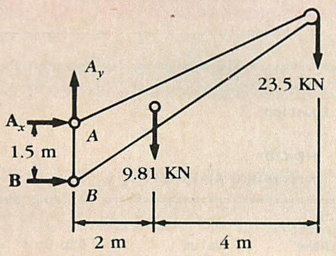
Victor E. Wright is manager of process engineering at Luckett & Farley in Kentucky. He has a degree in industrial management from the University of Cincinnati.

A SAMPLE PROBLEM

Stress Analysis. A fixed crane has a mass of 1,000 Kg (kilograms) and is used to lift a 2,400-Kg crate. It is held in place by a pin at A and a rocker at B. The crane's center of gravity is at G. Determine the reaction components at A and B.



Solution. A free-body diagram of the crane is drawn. Multiplying the masses of the crane and of the crate by g = 9.81 m/s² obtains the corresponding weights, that is, 9,810 N (newtons) or 9.81 KN (kilonewtons), and 23,500 N or 23.5 KN. The reaction at pin A is a force of unknown direction, represented by its components $\mathbf{A}_x$ and $\mathbf{A}_y$. The reaction of the rocker B is perpendicular to the rocker surface; thus it is horizontal. It is assumed that $\mathbf{A}_x$, $\mathbf{A}_y$, and $\mathbf{B}$ will act in the directions that are shown.



To determine B, the statement expresses first that the sum of the moments of all external forces about point A is zero. The equation obtained will contain neither A_x nor A_y because the moments of $\mathbf{A}_x$ and $\mathbf{A}_y$ about A are zero. Multiplying the magnitude of each force by its perpendicular distance from B:

$$+\circlearrowleft \Sigma M_A = 0: \quad +B(1.5 \text{ m}) - (9.81 \text{ KN})(2 \text{ m}) - (23.5 \text{ KN})(6 \text{ m}) = 0$$
$$B = +107.1 \text{ KN} \qquad \mathbf{B} = 107.1 \text{ KN} \rightarrow$$

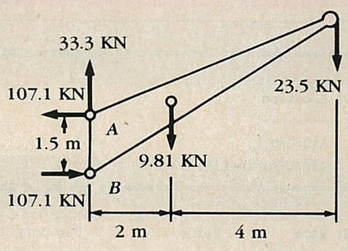
Because the result is positive, the reaction is directed as assumed. The magnitude of A_x is determined by expressing that the sum of the horizontal components of all external forces is zero:

$$\overset{+}{\rightarrow} \Sigma F_x = 0: \quad A_x + B = 0 \qquad A_x + 107.1 \text{ KN} = 0$$
$$A_x = -107.1 \text{ KN} \qquad \mathbf{A}_x = 107.1 \text{ KN} \leftarrow$$

The result is negative, so the sense of $\mathbf{A}_x$ is opposite to that assumed originally. In determining $\mathbf{A}_y$, the sum of the vertical components must also equal zero:

$$+\uparrow \Sigma F_y = 0: \quad A_y - 9.81 \text{ KN} - 23.5 \text{ KN} = 0$$
$$A_y = +33.3 \text{ KN} \qquad \mathbf{A}_y = 33.3 \text{ KN} \uparrow$$

Adding vectorially the components $\mathbf{A}_x$ and $\mathbf{A}_y$ reveals that the reaction at A is 112.2 KN at 17.3 degrees to the upper left (when 0 degrees is horizontally left on the diagram).



The values that were obtained for the reactions may be checked by recalling that the sum of the moments of all external forces about any point must be zero. For example, considering point B:

$$+\circlearrowleft \Sigma M_B = -(9.81 \text{ KN})(2 \text{ m}) - (23.5 \text{ KN})(6 \text{ m}) + (107.1 \text{ KN})(1.5 \text{ m}) = 0$$

Kg = kilograms *KN = kilonewtons* *m = meters*

Formula/One solves this problem in figure 3 on page 115.

This material was obtained from Vector Mechanics for Engineers *by Beer and Johnston (1977), courtesy of McGraw-Hill Book Company.*

FORMULA/ONE VERSUS TK!SOLVER

PC Tech Journal's review of TK!Solver in September 1985 came virtually on the heels of Lotus's acquisition of Software Arts (the creator of the program). Lotus formed a new division, apparently to support TK!Solver, but never released a Lotus version. It would serve no purpose to compare Formula/One and TK!Solver if that situation had not changed: Lotus has since sold TK!Solver to Universal Technical Systems (UTS). The enhanced and repackaged version 1.6 of TK!Solver is now on the market.

When Formula/One was introduced, it was much faster than the then-current version of TK!Solver, which was written is an obscure Software Arts language. Subsequent modifications have improved TK's speed; the most significant change has been compilation. Software Arts had included a machine-specific interpreter and a machine-independent program. The UTS version is a combination of C and assembly language—its speed is now on a par with Formula/One. As a benchmark, three equations

$$x + 3\cos h(y) + 7z = 5$$
$$e^x + y + z = 1$$
$$x - y = \cos(z)$$

were solved for x, y, and z with both programs. TK!Solver solved the equations in 40 percent less time than Formula/One (6.3 seconds versus 10.0 seconds on a PC/AT).

THE NEW TK!SOLVER

TK!Solver version 1.6 also adds several features that are similar to those of Formula/One: function key support, color displays, foreign character support, and full-word prompts.

At a more practical level, TK 1.6 adds an **if...then** rule form that greatly simplifies the construction of models that must incorporate a branching capability. With this construct, a user can write a rule in the form

If shape = circle then area = pi * radius²

Formula/One provides a similar capability in the form of the conditional functions **ifeq(...)**, **ifne(...)**, **ifge(...)**, **ifgt(...)**, **ifle(...)**, **iflt(...)**.

Formula/One and TK!Solver are based on essentially the same concept, but they approach modeling in slightly different ways. Formula/One provides a powerful set of built-in functions; TK does not. However, TK permits the definition of user functions. (TK2, highlighted in a sidebar with the TK!Solver review, promises procedural user functions.) Many of Formula/One's built-in functions can be duplicated with TK's list solver, although this process adds to development time. The introduction of procedural user functions should cut development time for equation-solving programs in general.

Consider Formula/One's built-in functions for evaluating polynomials. **Poly(x, list)** is the counterpart of the TK function of the same name, but **ipoly(x, list)** and **dpoly(x, list)** also are provided to evaluate the integral and derivative of the polynomial in x with coefficients stored in list. Duplicating these functions in a TK!Solver model requires the construction of a model within the model.

Even more attractive than **ipoly(...)** and **dpoly(...)** is the Formula/One **integ(fx,x,a,b,[n])** function, which evaluates the integral of the function fx on x, between the limits a and b, using Simpson's Rule. This procedure is a numerical method for finding the area under a curve (which is the definition of the integral of a function) by evaluating the function at a number of intervals and summing the products of the value of the function and the interval width. Simpson's Rule provides increasing accuracy with smaller intervals, which can be specified by the optional argument, n. (This function approximates integrals numerically and should not be confused with true symbolic integration, which neither program includes.) Nonetheless, **integ(...)** is certainly a worthwhile function, and many complex integrals can only be approximated, not solved exactly.

Business modelers will appreciate other Formula/One built-in functions, such as **cashf(x,y,z)**, which returns a cash flow per term, and **npv(rate, list)**, which returns the net present value of a list of cash flows. Of course, these functions can be duplicated in TK!Solver, but only at the expense of development time.

ITERATIVE SOLUTIONS

Formula/One and TK!Solver differ in program organization in several ways. For one thing, Formula/One provides two sheets for which TK has no coun-

FIGURE 1: *Iterative Solution*

```
============================== Equation Sheet ==========================

St Equation
-- --------
   A+B=C*D
   sin(C-A)=ln(E/F)
============================== Variable Sheet ==========================

St Name          Value           Dsp Unit  Cal Unit  Comments
-- ----          -----           --------  --------  --------
   A
I  B             3.4
   C
I  D             1.325
I  E             .567
I  F             .25
```

Formula/One and TK!Solver use iteration to find values of *A* and *C* that satisfy equations on the above sheet. TK!Solver requires initial guesses while Formula/One generates its own guesses if none are provided.

FIGURE 2: *Graphical Solution*

```
============================== Equation Sheet ==========================

St Equation
-- --------
*   A+B=C*D
*   error+sin(C-A)=ln(E/F)
============================== Variable Sheet ==========================

St Name          Value           Dsp Unit  Cal Unit  Comments
-- ----          -----           --------  --------  --------
OL A
I  B             3.4
IL C
I  D             1.325
OL error
I  E             .567
I  F             .25
```

Both programs offer an alternative: given a range of values for *C*, the solver computes values of *A* and *error*. Both packages can produce the plot in figure 3, indicating an infinite number of *C* values for which *error* is zero.

terparts: curve fit and regression analysis. These built-in capabilities save the modeler considerable time.

Formula/One and TK!Solver also differ in solution strategies, particularly their iterative solvers. Both use guess values for selected variables and solve the model repeatedly, comparing the results of each solution to those of the previous iteration. When the results of two iterations differ by a preset value, the program considers the model solved. Both programs use a comparison tolerance value to make the determination. In addition to this comparative value, however, TK uses a *typical value* to scale the comparison tolerance for use with very small or very large values.

Both programs require iterative solving under similar circumstances; for example, expressions that cannot be factored in the unknown variable must be solved iteratively. TK requires guess values to be input; Formula/One does not. The Formula/One manual does concede that guess values should be entered in some cases, most notably for models with multiple roots.

The figures point out other differences. Given the set of input values in figure 1, TK and Formula/One do not produce identical results, although both yield good approximations of the solution. TK requires that *C* be set to G(uess) status, while Formula/One sets both *A* and *C* to G automatically. On a PC/AT (6 MHz), Formula/One took about 10 seconds to solve the model in 12 iterations (and required that the solve command be given twice unless the maximum iteration count were raised beforehand). TK!Solver solved the model in 3 seconds with 5 iterations.

With Formula/One, only the input values need to be entered by the modeler. Variables *A* and *C* can be left blank: the program will produce guess values automatically for both variables, then it will invoke the iterative solver. Following the completion of the first 10 iterations (assuming the default of 10 has not been changed to a higher value), Formula/One displays a G status indicator for the two variables. If acceptable values are not found after 10 iterations, the equation status field indicates whether the values are D(iverging) or C(onverging). TK!-Solver, however, will not solve the model unless at least one variable—either *A* or *C*—is set to G.

In real-world engineering situations, the time required to set up the model is usually more significant than the time required for solving. For one thing, set-up requires both the computer and an operator, solving can be left unattended. For models that require iterative solutions, particularly those that have equations with multiple roots, set-up includes both developing the model in equation form, and discovering the best-guess values.

Formula/One requires more attention to guess values because it supplys a value for any variable not assigned an input or guess value by the modeler. This requirement complicates the process of discovering best-guess values. TK will converge to the correct set of roots given fewer (correct) best-guess values.

Both Formula/One and TK!Solver provide the appropriate tools to arrive at best guesses by graphical means. To that end, the equations in figure 1 are modified in figure 2.

By adding the variable *error* to the model, the user can remove the guesswork from the solution process. The idea is as follows: by plotting error against the unknown variable *C*, the user can pick a value of *C* for which error is zero; in effect, a solution of the equation. Leaving the *B*, *D*, and *F* values as they were, the user can create an input list for *C* and an empty (output) list for error. The user may leave *A* a single-valued variable or associate it with a list.

After these steps are taken, the model can be solved with the list solver, and error can be plotted as a function of *C*. Both TK!Solver and Formula/One produce a plot like the one in figure 3. The user can refer to the list sheets for error and *C* to locate conditions of minimum error.

Incidentally, this exercise points out another difference. Formula/One has no explicit command to invoke the list processor, but TK!Solver does. If a Formula/One model contains any references to lists in the variable sheet status column, it invokes the list solver. Input lists must be assigned IL status, and output lists must be assigned L. To return to a single solution mode, all list status indicators must be removed. With TK, variables can be associated with lists freely. The list processor is not invoked unless the /L command is entered.

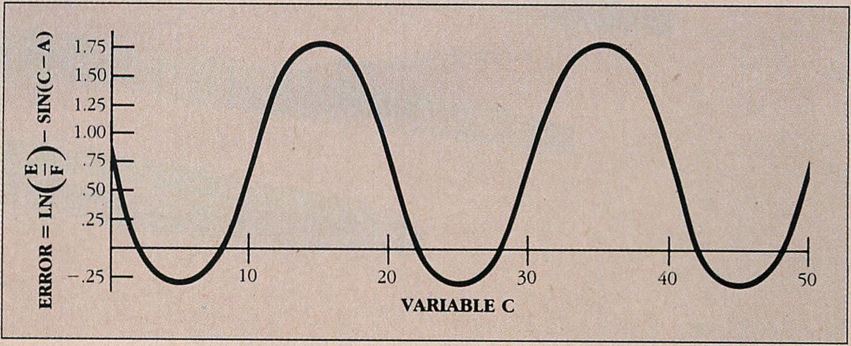
MODEL COMPETITION

Formula/One and TK!Solver were written to solve essentially the same types of models, and to meet the needs of the same group of users. Both perform as intended and produce accurate results when used correctly. The differences between the two are both cosmetic and functional. TK!Solver has the distinction of being the first of the genre (at least for microcomputers), but Formula/One cannot be called a TK!Solver clone.

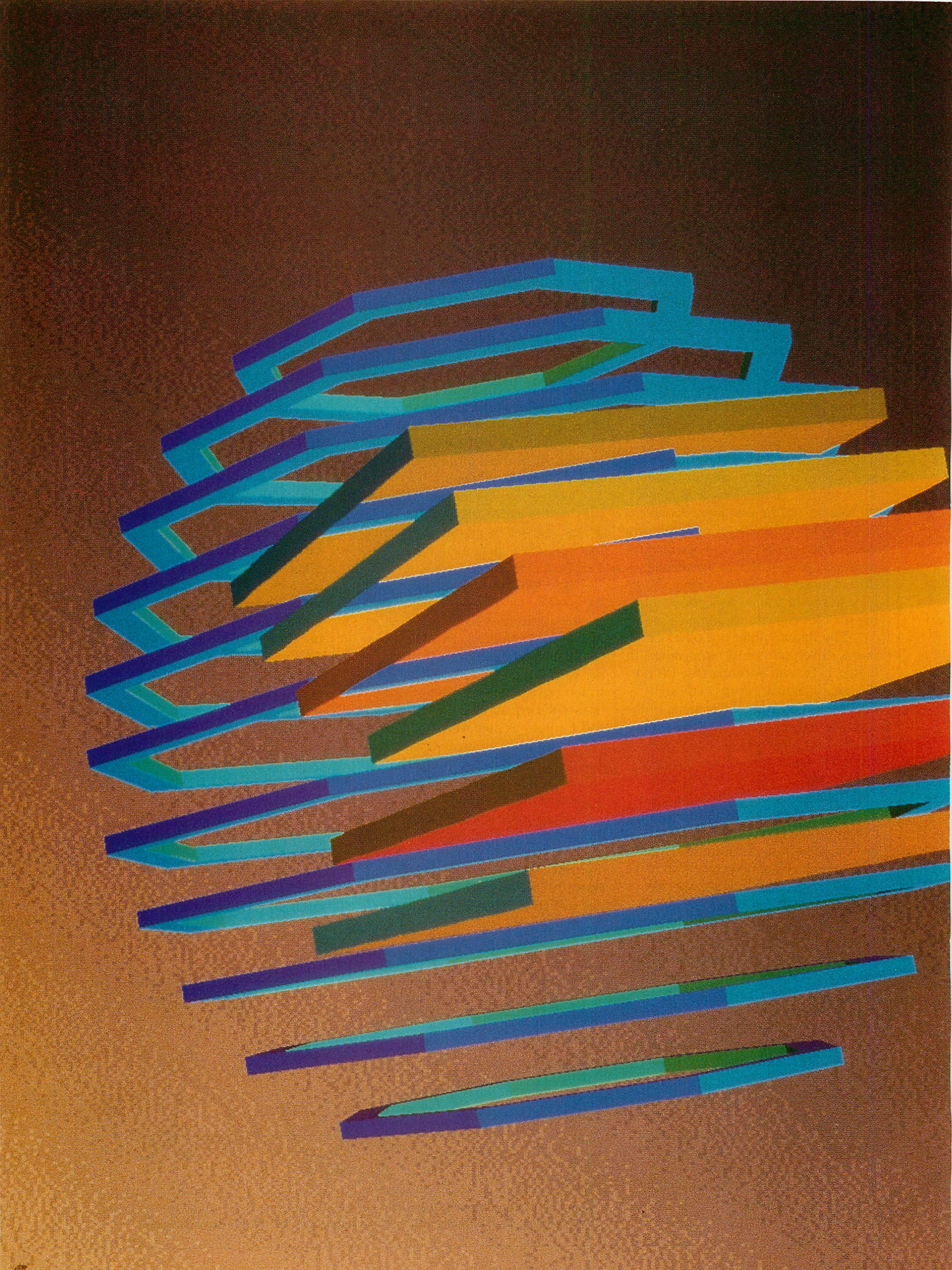
A model that can be solved on one of these two programs can be solved on the other (with varying degrees of difficulty). TK!Solver has the edge in flexibility, but Formula/One offers built-in functions.

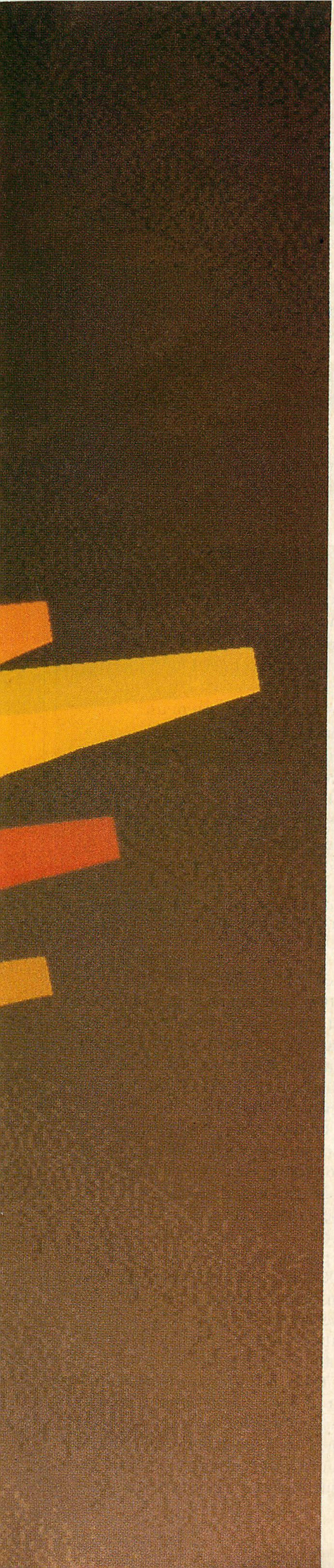
—*Victor E. Wright*

FIGURE 3: *Error as a Function of C*



A plot of the equations in figure 2 reveals that error is a sinusoidal function of C. To solve the equations exactly, pick a value of *C* for which error is zero.





PC Versions of iRMX

PC/RTX and AT/RTX from RTCS Corporation are IBM PC-specific implementations of Intel's ubiquitous, highly adaptable iRMX86.

RICHARD M. FOARD

Intel's iRMX86, the largest and most venerable of the 8086-family realtime operating systems, was designed in the late 1970s to address a broad range of computing needs. Conceived in a commercial world that had not yet seen the IBM PC, it was aimed primarily at large, high-volume systems manufacturers and integrators. Devised to support the 8086 family in the widest possible range of hardware environments and application settings, iRMX was built to be highly configurable with respect to its hardware environment.

Real-Time Computer Science Corporation (RTCS), working under an Intel resale license, has taken advantage of the system's configurability to create iRMX-based operating systems for two of the world's most common vehicles for 8086 family processors: PC/RTX for the IBM PC and AT/RTX for the PC/AT. These two products, which incorporate Intel's release 6.0 of iRMX86, are potentially attractive to both high- and low-volume resellers of realtime systems based on standard PC hardware. High-volume resellers can build upon the configuration work RTCS has done for the PC and AT hardware sets. Low-volume resellers, in addition to avoiding the chore of configuring iRMX, can enjoy more favorable low-quantity pricing from RTCS than they would see if they purchased iRMX directly from Intel.

At $995 for PC/RTX and $1,495 for AT/RTX, these systems are not priced or packaged for the casual experimenter. Most existing RTX users are larger corporations that already held iRMX licenses from Intel when they purchased RTX from RTCS. RTCS's focus on larger customers with technical sophistication is evident in its handling of product documentation: RTX is distributed with a 24-page, RTCS-authored installation

COMPUTER GRAPHIC • DOV JACOBSON

guide and Intel's weighty *Introduction and Operator's Reference Manual.* Documentation on program interfaces to iRMX is not distributed with RTX. Customers not already holding the documents by virtue of holding iRMX licenses from Intel are referred directly to Intel for the two volumes of the *iRMX Programmer's Reference Manual.*

Resellers building systems for nonstandard PC/AT hardware sets can purchase configurable (but more costly) variants of RTX from RTCS. These products, unlike their PC and AT counterparts, pass through all iRMX configurability, but require a direct license agreement with both Intel and RTCS.

TABLE 1: *UDI System Calls*

CALL	FUNCTION
DQALLOCATE	Allocates memory block
DQATTACH	Establishes a file connection
DQCHANGEACCESS	Changes file access rights
DQCHANGEEXTENSION	Changes file name extension
DQCLOSE	Closes a file
DQCREATE	Creates a file (and file connection)
DQDECODEEXCEPTION	Translates exception code
DQDECODETIME	Returns current date and time
DQDELETE	Deletes a file
DQDETACH	Detaches a file connection
DQEXIT	Returns control to RTCS/UDI (PC-DOS)
DQFILEINFO	Returns file information
DQFREE	Frees a memory block
DQGETARGUMENT	Gets a command line argument
DQGETCONNECTIONSTATUS	Returns file connection information
DQGETEXCEPTIONHANDLER	Returns current exception handler address
DQGETSIZE	Returns the size of a memory segment
DQGETSYSTEMID	Returns system identification string
DQGETTIME	Gets current date and time
DQOPEN	Opens a file connection
DQOVERLAY	Loads an overlay
DQREAD	Reads from a file connection
DQRENAME	Renames a file
DQRESERVEIOMEMORY	Reserves memory for system use
DQSEEK	Sets/changes I/O position in file
DQSPECIAL	Sets console input treatment
DQSWITCHBUFFER	Switches parsing buffer
DQTRAPCC	Sets Ctrl-C handler
DQTRAPEXCEPTION	Sets hardware-detected exception handler
DQTRUNCATE	Truncates a file
DQWRITE	Writes to a file connection

By using RTCS/UDI running under PC-DOS, the developer can use the same calls to the service routines as a developer who is using the VDI running under RTX itself.

iRMX86/RTX COMPONENTS

In the spectrum of possible iRMX configurations, RTX lies at the "fully loaded" end. The RTX products contain all of the iRMX system's core and optional components.

At the heart of RTX lies the iRMX Nucleus. The Nucleus controls system resources and coordinates tasking and interrupt processing. Layered atop the Nucleus are the iRMX basic and extended I/O systems—RTX BIOS and EIOS—which provide device-independent, file-based input and output. The I/O systems manage hierarchical file systems on random access devices and provide access to simpler devices. The iRMX Universal Development Interface layer (UDI), a standard set of service routines, allows RTX to host Intel's language compilers and other software development tools. (UDI system calls are listed in table 1.) The iRMX Human Interface component provides a set of command line interpretation and operator/program communications services that allow a system's users to invoke programs and control system operation. RTX also incorporates the iRMX Bootstrap and Application Loaders.

In the hierarchy of the system, beginning with the application system and continuing with UDI, EIOS, RTX BIOS, and the Nucleus, each element can communicate directly with those elements beneath it (see figure 1). For example, the application system can talk to all of the other elements, but the UDI can talk directly only to the EIOS, the RTX BIOS, and the Nucleus.

The increased power of Intel's 8086 family and other late-1970s microprocessors made an expanded complexity in operating software architectures necessary. Integrated circuit manufacturers, anticipating their customers' difficulties in grappling with growing levels of complexity, dramatically boosted their efforts to provide systems level software as part of their product packages. In Intel's case, concerted efforts to support the 8086 family yielded a more highly structured, more formalized approach to operating system design than had been seen previously in commercial microcomputing. The development of iRMX was an effort more akin to that which produced UNIX than to that which produced CP/M or PC-DOS. These efforts culminated in 1980 with the introduction of the layered, object-oriented iRMX86 operating system.

Aside from its substantial size, it is iRMX's object-oriented design that sets it apart from the microprocessor operating systems that preceded it: iRMX can be described completely in terms of the properties and behavior of abstract *objects* of various types. Objects are tasks, units of resources used by tasks (such as memory segments), special entities used in intertask communication, or structured compositions of other objects. The iRMX Nucleus allows the creation, deletion, and manipulation of eight types of objects: jobs, tasks, segments, semaphores, mailboxes, regions, extension objects, and composite objects. All of its interfaces are described in terms of these basic object types.

Of the eight basic types, tasks (and composite objects that include tasks) are the only active objects. iRMX terminology uses *task* in the conventional sense: tasks perform the work of a system by creating, deleting, manipulating, and passing other objects around. When iRMX objects are passed among tasks or from tasks to system calls, they do not move physically from one place to another. Instead, a compact (16-bit) identifier called a *token* is passed.

Every iRMX system is structured as a set of one or more *job* objects. Jobs provide a way for a systems designer to achieve close control over the distribution of system resources. Jobs are most frequently used when iRMX provides

Read what the experts say about the PC's Limited AT.

"I have no qualms recommending this system to experienced users who need the added speed."
—Sol Libes,
Micro/Systems Journal,
January/February 1986

"The PC's Limited AT™ proved to be the functional equivalent to the IBM® product, only faster. Priced at roughly one-third less, it rates as one of the best bargains available."
—Winn L. Rosch,
PC Magazine,
February 25, 1986

"If price is your primary consideration, skip both AT&T® and TI® and choose this issue's best buy, the PC's Limited AT™. Check its benchmark results—you won't be sacrificing performance to save money."
—*PC Magazine,*
Editor's Choice,
February 25, 1986

30 DAY Money-Back Guarantee

And if that's not enough to convince you, now we've got a 30-day money-back guarantee.

- One Year Limited Warranty
- 80286-based System Unit (Runs at 6MHZ. 8MHZ option included at no extra charge.)
- 1024K on Mother Board
- 1.2 Meg Floppy Drive
- Combined Floppy/Hard Drive Controller Card
- AT Keyboard
- 192W Power Supply
- 2 Serials and 1 Parallel Port
- Clock/Calendar with Battery Backup
- Operations Manual

$1995

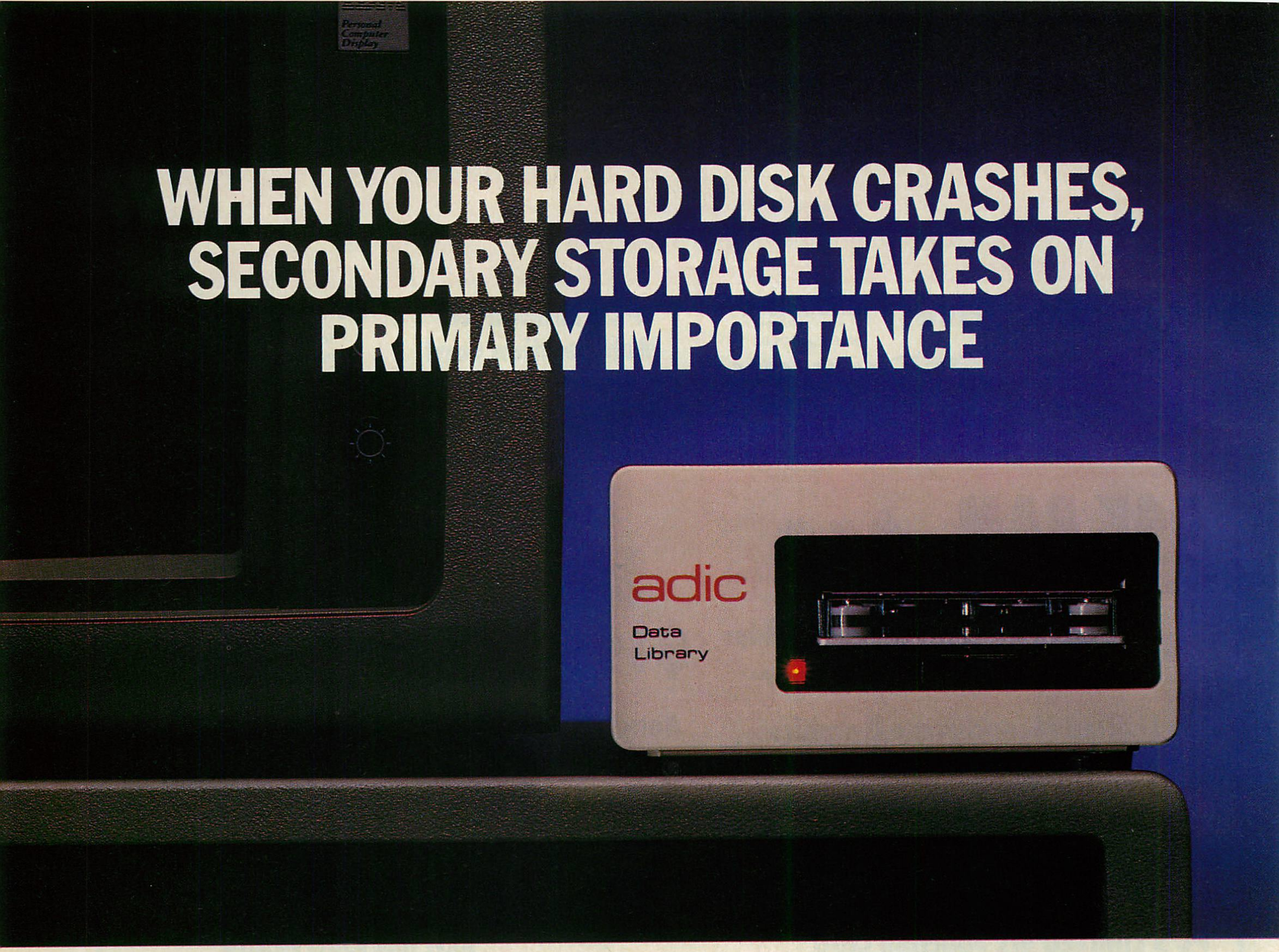
Microsoft GW Basic™ $95 • IBM DOS 3.1 $85 • 80287 $195

I'm Michael Dell, President of PC's Limited. And I want you to know about your latest industry-leading step. We're so sure that you'll like everything about our PC's Limited AT that we're willing to offer you a 30-day money back guarantee on it. This machine, with 1024K on the Mother Board and an 8MHZ option included at no extra cost, has been a standout since we introduced it. So give us a chance to show you what it can do for your operation. With a deal like this, there's no way you can lose. (And see the opposite side of this page for information about our Turbo PC™, plus details on our guarantee.)

Michael Dell

CIRCLE NO. 145 ON READER SERVICE CARD

WHEN YOUR HARD DISK CRASHES, SECONDARY STORAGE TAKES ON PRIMARY IMPORTANCE



INTRODUCING THE TD440 TAPE BACKUP SYSTEM

What will you do when your hard disk fails? Too many people wait until that catastrophic moment to consider the importance of secondary storage.

And too many people rely on clumsy floppy disks or second rate tape systems to protect literally thousands of dollars of programs, data and time.

ADIC offers a better way—the TD440 Tape Backup System—a complete 40 megabyte ¼" tape backup system for IBM PCs, XTs, ATs and compatibles.

Available in standalone PC or network versions (IBM, Novell and 3-Com to name a few), the TD4400 solves virtually any backup need.

TOTAL DATA SECURITY

If you want to sleep nights without worrying about valuable data, the TD440 is for you. It's the newest in our line of superior quality tape systems, about which *PC Tech Journal* says, "It's hard to imagine a sturdier, more secure system…the army tank of tape drives," (Nov. 85).

Our QIC-100 tape format and redundant data recording error detection/correction techniques insure total data integrity. Even if a data error occurs, the TD440 automatically corrects it.

HOW TO RUN PROGRAMS FROM A DEAD HARD DISK…

You can't. Instead, when your hard disk crashes, simply switch over to the TD440. It acts just like a hard disk, responding to all DOS commands and lets you run programs directly from tape.

The TD440 keeps you up and running productively even while the hard disk is in the shop. Now that's true secondary storage!

BUY THE BEST—FOR LESS

At only $1490, the security you buy with the TD440 is a bargain. Ask the thousands of satisfied customers who have put their trust in ADIC—trust that is well-deserved.

Every TD440 is manufactured to exacting standards, using 3M drives and tapes in conjunction with ADIC technology. And our advanced backup and archival software makes data management a snap.

So if you want superior protection for your valuable data, call ADIC today at **1-800-336-1233**. The TD440—it's the tape backup system for people who refuse to accept second rate protection.

The TD440 tape backup system stores 40 Mbytes of data on easily transportable 3M DC2000 cartridges.

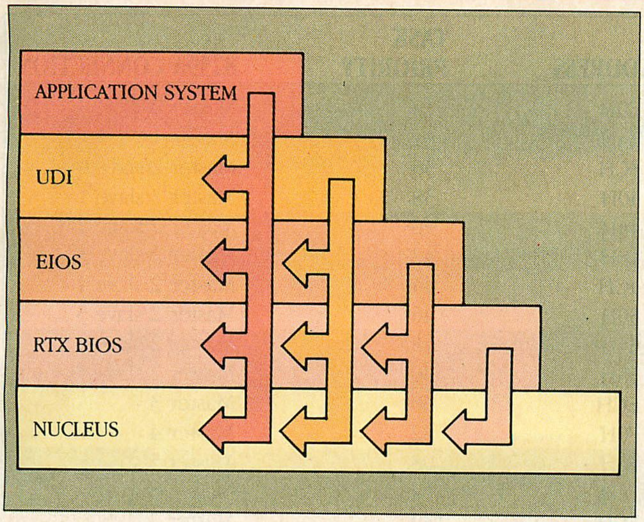
data cartridge technology

adic

Advanced Digital Information Corporation
P.O. Box 2996, 10201 Willows Road N.E.
Redmond, WA 98073-2996
(800) 336-1233, (206) 881-8004

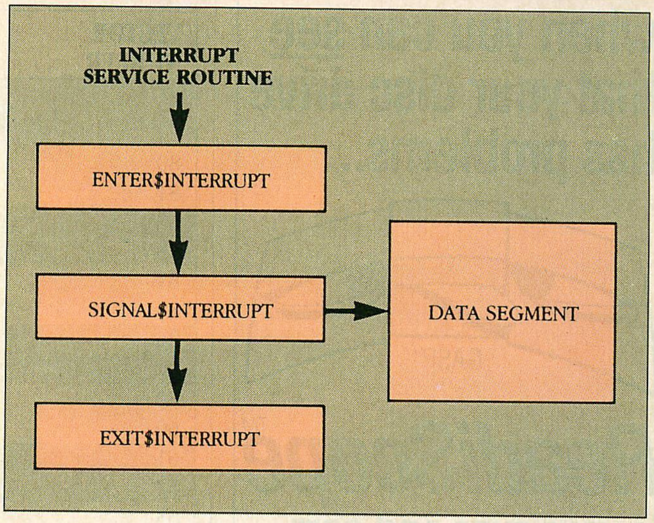
CIRCLE NO. 102 ON READER SERVICE CARD

FIGURE 1: *Hierarchy of Communication*



The hierarchy of the system allows any element to communicate directly with tasks beneath it in the tree; elements may not talk to higher-status elements.

FIGURE 2: *Interrupt Management*



The handling of some interrupt service routines is done by making an interrupt task associated with it. The ISR generates the data segment used later by the interrupt task.

the foundation for systems serving competing, interactive users. Systems dedicated to noninteractive applications are typically (though not necessarily) constructed entirely within a single job.

Jobs are groupings of tasks and other logically related objects. A system's jobs are linked into a tree structure through which resources are allocated among the system's tasks. Every iRMX system is structured as a tree of jobs descending from an initial root job. Each task is associated with a particular job and must obtain the resources it needs from its associated job or from one of its ancestors in the job tree. Tasks also may create new jobs; each new job is a descendant of its creating task's job in the job tree.

Every job carries with it a memory pool and a directory of the objects it contains. Various limits are specified when a job is created that remain in force throughout its life: a maximum size for its memory pool, a maximum number of entries in its object table, and so on. A job's limits apply to the job and to all its descendant jobs.

The role played by the job hierarchy in partitioning system resources is evident in iRMX's memory management facilities. When first created, jobs do not receive their entire memory pools. Instead, they are assigned only a designated minimum amount, and additional memory is allowed to migrate down the job tree as required to satisfy runtime requests for memory allocation.

Tasks obtain memory by making the system call CREATE$SEGMENT to create and allocate *segment* objects. Segments are paragraph-aligned parcels of memory that consist of enough contiguous paragraphs of physical memory to meet a task's request. When a task attempts to create a segment requiring more memory than is available in its job's memory pool, iRMX tries to "borrow" memory from the parent job. If the parent job cannot help, attempts to borrow continue to grandparents and beyond, all the way to the root job if necessary. A job retains memory it has borrowed until it is deleted, at which time all the memory its tasks have accumulated is returned to its parent job.

SCHEDULING AND TASKING

Each job is created with one initial task. A task that creates a new job optionally may pass it a token for a parameter object. This generality with respect to a job's parameters is characteristic of the iRMX brand of object-oriented design: the operating system does not presume to know the internal structure, size, or meaning of a job. The parameter object could be a memory segment containing operating instructions, a mailbox to be managed by the new job, or even another job object. A job's initial task typically performs initialization, then creates the required objects within the job, including other tasks.

iRMX maintains a preemptive, priority-based multitasking environment. Each task's context consists of its stack, its containing job, and a task-specific exception handling apparatus.

iRMX tasks may be assigned any priority level from 0 (the highest) to 255. The task priority scheme is intimately connected with the iRMX interrupt-handling support mechanism. Regardless of the number of tasks or interrupting devices present in a particular system, iRMX associates each even-numbered task priority between 4 and 130 with an external device interrupt; tasks running at these priorities are considered to be interrupt tasks. Priorities 131 through 255 are used for noninterrupt tasks. When an interrupt task runs, hardware interrupts associated with lower-priority interrupt tasks are automatically disabled. iRMX makes the interrupt-to-task priority association by assuming a fixed relationship between interrupt task priorities and locations in the processor's interrupt vector (shown for PC and PC/AT hardware configurations in table 2).

Basic task management for tasks not engaged in interrupt processing or intertask synchronization is accomplished using five system calls that move tasks among various *states*. CREATE$TASK and DELETE$TASK add and remove tasks from the system. Once created, a task contends normally with other tasks for processor time unless it enters the *sleeping* state, the *suspended* state, or both. A task sleeps when it makes a SLEEP system call to remove itself from contention for the processor for a specified length of time. Tasks become suspended when they are named in a SUSPEND$TASK system call; they remain in this state until named in a RESUME$TASK call. (A task that is suspended more than once must be named in a corresponding number of RESUME$TASK calls before it begins

TABLE 2: *Interrupt Procedure Relationships*

VECTOR NUMBER	ADDRESS	TASK PRIORITY	8259A CONNECTION
56	E0H	18	Master 0
57	E4H	34	Master 1
80	13CH	36	Master 2/slave 0
81	140H	38	Master 2/slave 1
82	144H	40	Master 2/slave 2
83	148H	42	Master 2/slave 3
84	14CH	44	Master 2/slave 4
85	150H	46	Master 2/slave 5
86	154H	48	Master 2/slave 6
87	158H	50	Master 2/slave 7
59	ECH	66	Master 3
60	F0H	82	Master 4
61	F4H	98	Master 5
62	F8H	114	Master 6
63	FCH	130	Master 7

Each location in the interrupt vector is automatically associated by RTX with an interrupt task priority and an 8259A connection. The IBM PC has only a master controller and no slave controller, providing 15 interrupt priority levels.

running normally again; iRMX tracks each task's suspension depth.) If a task is suspended while sleeping, it remains inactive until *all* blocking conditions have been cleared.

iRMX provides tasks with a number of informative system calls. Using the GET$TASK$TOKEN system call, a task can learn about itself from iRMX. In making this call, a task can indicate that it wants a token for itself, its job, its job's parameter object, or the system's root job. Using the GET$PRIORITY system call, a task can discover its own priority or that of any other task for which it holds an identifying token.

EXCEPTION HANDLING

The notion of *exception* is central to the iRMX rules of operation. Exceptions, for the most part, occur during system calls, in the form of I/O errors, improper call parameters, failures to allocate adequate essential memory.

Tasks and system calls react to the detection of exceptional conditions by calling on iRMX's highly structured exception-handling mechanism. In designing a task, a designer chooses an exception-handling method. First, he must choose when, if ever, a task's exception handler is to be invoked, which is done by setting the task's exception mode. A choice of the in-line exception-handling mode indicates that a task has *no* special exception handler. This is the familiar case in which the task itself is responsible for testing the outcome of every system call and processing errors if they arise. Alternatively, the designer may specify that a designated exception handler be invoked upon programmer errors (improperly made system calls, for example), environmental errors (such as unexpected end-of-file when reading from disk), or both.

If a task does not call on SET$EXCEPTION$HANDLER to set its own exception mode and handler, it receives by default the exception handling apparatus of its containing job. If no mode was specified when the job was created, the job and its tasks inherit a default, iRMX-supplied exception handling strategy that simply suspends or deletes any task causing an exception of any kind. Tasks can discover which exception apparatus they have established (or inherited) by using the GET$EXCEPTION$HANDLER call.

When iRMX invokes an exception handler, it passes to the handler codes that describe the exception and, in the case of exceptions arising from improperly used system calls, an indication of which parameter to the system call was in error. The systems designer can choose whatever strategy he deems appropriate for dealing with exceptions. In extreme cases, he may choose to eliminate the offending task entirely, by deleting it. Under other circumstances, he might log the error, ignore it, or attempt to correct the conditions that led to the exception. He also has the option, which is especially attractive during system development and testing, of establishing the iRMX debugger as the exception-handling routine for either some or all system tasks.

More Answers from CXI. #12.

How to get mainframe data to your remote PCs.

PCOX 3270 remote connections.

Now if your PCs are located out of town, they needn't be out of touch.

Because not only does CXI have more remote micro-to-mainframe connections to choose from; we have more cost-effective ones as well.

For example, all our remote products connect via modems. And emulate IBM® 3274 cluster controllers, 3278/79 terminals, 3287 printers and 3270 PCs.

So you can forgo the cost of that equipment when you link up with ours.

Take our PCOX/STANDARD REMOTE™ connection, for one. It gives you direct mainframe access from a remote PC.

Or choose PCOX/PLUS REMOTE.™ It can access one host session, one PC session and two electronic notepads—concurrently. And makes file transfer quick and easy.

Or, for multi-session performance, there's our PCOX/3270 PC REMOTE.™ It lets you work with up to five host sessions, two notepads and a PC session.

What's more, just by adding new software, you can add new features like our remote gateway for LANs.

So for the name of your nearest CXI distributor, call **(800) 225-PCOX**. In California, call (415) 424-0700. Telex: 821945.

Or send us the coupon. And let us turn these remote possibilities into reality.

For a brochure and more information on our PCOX 3270 remote connections, fill out this coupon or attach your business card and mail to CXI, Inc., 3606 West Bayshore Road, Palo Alto, CA 94303.

Name, Title ______

Company ______

Address ______

City, State, Zip ______

Phone ( ) ______

CXI

More micro-to-mainframe answers.

PCTJ6/86

CIRCLE NO. 217 ON READER SERVICE CARD

SYNCHRONIZATION

All intertask synchronization under iRMX is modeled in terms of tasks exchanging or sharing something. Exchanges are accomplished using objects of type semaphore and mailbox; sharing is controlled using regions. Like other objects, exchange objects and regions may be dynamically created or destroyed, as required.

Semaphore objects implement the well-known, general-purpose Dijkstra synchronization primitives. An application creates a semaphore as the custodian of some number of abstract units, usually corresponding to events or resource units in the application domain. Tasks send units to a semaphore (signal or post) using the SEND$UNITS system call; they request units from a semaphore (wait) by calling RECEIVE$UNITS.

Tasks calling RECEIVE$UNITS specify the number of units they wish to receive. If all the requested units are not present in the semaphore (or if other tasks are waiting ahead of the caller) the calling task is queued up to wait, either on a first-come, first-served basis or by task priority relative to other waiting tasks. Queuing treatment is determined by a configuration option that is set at the time a semaphore is created.

In addition to the number of units they seek, calling tasks specify the amount of time they are willing to wait if their request cannot be satisfied immediately. The specified waiting time can range from zero (give up immediately if the requested units are not there) to infinity (wait as long as necessary to get them).

Mailbox objects are used to mediate the transfer of messages across task boundaries; tasks obtain access to mailboxes by calling SEND$MESSAGE and RECEIVE$MESSAGE. Instead of a semaphore's abstract units, tasks communicating via mailboxes send and receive messages that can be iRMX objects of any type. Each mailbox contains an object queue into which messages sent when no receiving task is waiting are placed. Object queues are managed on a strict FIFO basis.

As with semaphores, a task that creates a mailbox specifies whether tasks waiting at mailboxes are to be queued by arrival order or by task priority. Tasks attempting to receive messages have the same options that those receiving semaphore units have: they can choose whether or not to wait at an empty mailbox and, if they choose to wait, for how long.

iRMX maintains object queues in two parts: high-performance and overflow. The size of a queue's high-performance part is specified by the creating task at the time a mailbox is created. Messages passed through a mailbox while its object queue is small enough to be contained entirely in the high-performance part are processed quickly because they move through memory that is permanently allocated in association with the mailbox. When an object queue overflows its high-performance part, iRMX allocates additional memory from the mailbox's containing job to accommodate the overflow. Overflow processing slows message-passing down somewhat, but allows message queues to grow until all a job's memory is exhausted. The high-performance/overflow structure of object queues relieves the designer of the responsibility for establishing a fixed maximum size for each of a system's inter-task data queues.

Applications designers can exploit iRMX's fully general treatment of inter-task data flows through mailboxes. Because any object's token can be sent via a mailbox, the same mailbox can be used to pass memory segments, jobs, or even other mailboxes from one task to another. After a particular RECEIVE$MESSAGE call, a task receiving a message from such a mailbox can use iRMX object management facilities to determine the type of object it has received and tailor processing accordingly.

Although iRMX semaphore objects can meet any synchronization requirement that might arise in an application, iRMX provides another standard synchronization object—the *region* object. Tailored to perform the synchronization activities required when two or more tasks share a data area, region objects are useful when two or more concurrently executing tasks share a data area that may be modified by one or more of the cooperating tasks.

Synchronization is required when tasks share a common data area, to prevent the reading of inconsistent, corrupt information that can result when one task changes a data area while another is in the midst of reading it. The situation is avoided by coding each task that reads or writes a shared area in order to bracket its use of the area with system calls to gain (RECEIVE$CONTROL) and release (SEND$CONTROL) control of the area's associated region object. Used in this way, regions guarantee mutual exclusion and eliminate the possibility of tasks reading corrupted data.

iRMX designers chose to provide regions as well as semaphores because, although they can be used to accom-

For you, good is just not good enough.

For serious programmers, good isn't good enough. You need the best ... the best tools, the best advice and the best support.

At Lifeboat, we've been selling to programmers since 1976, so we know quality when we see it. And we're committed to a full-service program that goes beyond just selling you the best software at competitive prices. Our expert staff can help you decide which programs are best for your needs and provide you with all the technical support you may require. You can rely on Lifeboat for the complete solution to your programming needs.

LANGUAGES

Lattice C *New 3.0 Version*
The best selling C Compiler has been upgraded to give you more functions and features. Lattice C 3.0 contains 200 new library functions, better code generation, support for new data types (void, enum, unsigned char, unsigned long), support for the 80186/80286 instruction set and the ability to generate in-line 8087/80287 instructions. Lattice C is the C Compiler for professional developers.

RUN/C—The C Interpreter *Upgraded Version*
Learn C the natural way with RUN/C. The user interface is similar to BASIC with easy familiar commands. The new 2.0 version of RUN/C comes with a full-screen editor and other enhancements.

RUN/C Professional *New*
All RUN/C's capabilities plus powerful features for program development. Load your favorite object libraries with RUN/C Professional. Contains a full-screen editor and source code debugging facilities.

Mark Williams C Programming System
A complete C development environment.

Pro Pascal
A truly standard Pascal. Produces fast, tight code with plenty of compile time options and simple one-line commands.

BetterBASIC *New Version*
Now you can program in BASIC and use the full memory of your PC, create structured programs using functions and procedures, make your own library modules and more. Now compatible with Microsoft BASIC.

LANGUAGE UTILITIES

Plink86 Plus
An overlay linkage editor for linking 8086/8088 object modules. Supports an unlimited size file, unlimited number of modules and up to 4095 hierarchical overlays stacked as many as 32 levels deep. Plink86 Plus contains new features for memory caching, library allocation, file merging and overlay reloading.

Pfix86 Plus
A symbolic and source level advanced debugger for programming professionals.

C-SPRITE
Symbolic debugging at both source level and machine level. Monitor your program by setting breakpoints, examining registers and variables, or single-stepping through code. Handle a set of commands as a macro. C-SPRITE and the Lattice C Compiler are a natural pair for program development and debugging.

BASTOC
A BASIC to C translator for the BASIC programmer who wants to upgrade to C.

EDITORS

LSE
A powerful yet inexpensive full-screen editor designed for efficiency and ease of use. Its speed is optimized by writing directly to video memory. The Lattice Screen Editor provides a multi-window environment, ability to reprogram keys, support for keyboard macros, an undo command, an error tracking mode, on-line help and more.

VEDIT Plus
A full-screen text editor for program development and word processing. It contains powerful features including use of macros, on-line help facility, paragraph formatting, and file comparison.

Pmate
The programmer's editor with an extensive macro command language. Compile in the background while you continue to edit files.

EMACS
Customizable editor including windowing, multi-tasking and special modes for C and Pascal.

FUNCTIONS

C-Food Smorgasbord
Library of time saving utility functions including a BCD package, an IBM PC BIOS interface, level 0 I/O functions, a terminal independence package and more.

Essential C Utility Library
Over 300 functions, with special attention given to screen handling, windows and business graphics. Source code is included.

PforCe *Brand New*
An optimized library of an impressive 400 plus functions and subsystems. Included is a window management system with overlapping or tiled windows and a database system with B-tree file structure. Several pre-coded screens including "Lotus" style are supplied. And there's much more! Routines for interrupt driven communications, background tasks, and string/table parsing, along with functions for field/screen editing and validation are all part of this superbly written and documented software. Complete source code is included.

The Greenleaf Comm Library
A library of over 120 communication routines. Contains functions to create interrupt driven routines or perform direct I/O to multiple Comm Ports. Its strengths are in asynchronous communications, interrupt mode, modem control, XMODEM, XON/XOFF and flow control. Interfaces with Lattice, Microsoft, Wizard, Desmet, CI86, Aztec and Mark Williams C compilers.

The Greenleaf Functions
A mature library of over 200 functions. Version 3.0 offers all new indexed documentation, with an abundance of examples. Source code included.

GRAPHICS and SCREEN DESIGN

GSS*CGI GRAPHICS *New*
The GSS Computer Graphics Interface is designed for creating high performance graphics-based applications. GSS*CGI speeds up application development and provides compatibility with a wide range of peripherals. It's the only CGI implementation that provides true device-independence for both raster and vector graphics. Language bindings are available to support development of GSS*CGI based applications using Lattice C, Microsoft C, FORTRAN, Pascal, BASIC Compiler and Macro Assembler. Products in the GSS*CGI GRAPHICS line include the GSS Graphics Development Toolkit, the GSS Kernel System, the GSS Plotting System and the GSS Metafile Interpreter.

Essential Graphics *New*
A brand new graphics library for C programmers with the emphasis placed on ease of use and portability. No royalties.

Multi-Halo
Library of over 200 graphics functions, supporting all of the popular graphics boards.

Panel
A powerful tool for interactive screen design.

For more information on these and other products in our complete line call:

1-800-847-7078 In NY: 914-332-1875

LIFEBOAT

The Full-Service Source for Programming Software.

INTERNATIONAL SALES OFFICES

Australia:
Fagan Microprocessor Assoc.
Phone: (61) 3699-9899
Canada: Scantel Systems
Phone: (416) 449-9252
England: Grey Matter, Ltd.
Phone: (44) 364-53499
Italy: Lifeboat Assoc., S.p.A.
Phone: (02) 656-841
Japan: Lifeboat Japan
Phone: (03) 293-4711
Spain: Micronet, S.A.
Phone: (34) 1-457-5056
The Netherlands:
GIGA Computer Products
Phone: (31) 10-771846
SCOS Automation BV
Phone: (31) 20-106922

Lifeboat
55 South Broadway
Tarrytown, NY 10591

The names of products listed are generally the trademarks of the sources of the products.

CIRCLE NO. 144 ON READER SERVICE CARD

plish mutual exclusion, semaphores leave a system vulnerable to some potentially harmful effects. For example, a task that obtains control of a resource by consuming a semaphore's single unit can tie the resource up forever if it is suspended or deleted while still holding control of the unit. This cannot happen with regions because iRMX automatically defers any attempt to suspend or delete a task until it releases any regions it holds.

Regions also address a more subtle weakness that arises from the use of semaphores for mutual exclusion. Using semaphores, a high-priority task that requests control of a resource held by a low-priority task must wait until the low-priority task releases the semaphore. This delay is inherent in mutual exclusion and is unavoidable. If a task of intermediate priority preempts the low-priority task, however, the waiting high-priority task must sit idle until the intermediate task suspends and allows the low-priority task to continue executing. This additional delay can be considered excessive because it needlessly holds up the high-priority task while the intermediate task continues in its unrelated processing.

iRMX eliminates delays of this sort when regions are used by automatically elevating the priority of a task holding a region to equal that of the highest priority task waiting for the region. If another task arrives, the priority is raised again. When a task releases a region, its original priority level is automatically restored.

Regions can be configured (at the time they are created) to queue waiting tasks in either arrival order or task-priority order. Tasks requesting control of a region can do so in one of two ways. They can call ACCEPT$CONTROL either to gain control or to receive immediate notification that the region is busy, or they can call RECEIVE$-CONTROL, which waits as long as necessary to gain control of a region. Unlike its semaphore-manipulating counterpart, RECEIVE$CONTROL does not allow the calling task to specify a time interval after which it is to give up.

INTERRUPT MANAGEMENT

From a software architect's point of view, the IBM PC's 8259A interrupt controller device (or pair of devices in the AT's case) provides two important interrupt management services. First, as configured by iRMX, it imposes a priority ordering on the PC's interrupting devices. Second, it allows the selective masking of any hardware interrupt.

A device's interrupt priority, or *level* in Intel parlance, determines the treatment its interrupt service routine (ISR) gets when interrupted by another device before its processing of a first interrupt is complete. If the newly interrupting device lies at a higher priority level than the running ISR's device, a nested interrupt is allowed to occur, and control is vectored immediately to the new device's ISR. If it lies at an equal or lower priority level, the second interrupt is deferred until the first ISR issues an end-of-interrupt (EOI) instruction to the interrupt controller.

Most PC realtime systems are built to cooperate with the PC's standard approach to nested interrupt management (those that switch to a special interrupt stack, for example, must keep track of interrupt nesting so they do not switch to the stack when it is in use). Most do *not* rely upon or make use of the 8259A's ability to mask off specific interrupts. As a consequence, most realtime systems accept interrupts any time a task is running, in effect giving every ISR priority over the system's highest

priority task. iRMX is unique in that it uses the 8259A's interrupt masking capability to place task priorities and (hardware) interrupt levels in the same priority space. Under iRMX, it is possible to run tasks at priority levels high enough to take precedence over some or all hardware interrupt processing; a running high-priority task can block the execution of lower-priority ISRs.

In designing an interrupt service procedure for a multitasking system, a programmer must choose among several strategies. If the device's interrupts can be dismissed quickly, the job of servicing the device can be performed within an interrupt service routine (ISR). In this case, the device's ISR contends for processing time only with the ISRs of higher-priority devices. If more time-consuming work is required, or if potentially suspending system calls must be made in the course of processing an interrupt, it is best to code a minimal ISR that runs just long enough to signal a task, and allow the task to complete the rest of the processing.

iRMX can provide comprehensive support for interrupt-handling tasks by employing the 8259A interrupt controllers' ability selectively to mask interrupt levels. Tasks that identify themselves to iRMX as *interrupt tasks* get special treatment from the scheduler: when an interrupt task is activated by its companion ISR, the interrupts of all lower priority devices are automatically disabled and are held off until the task has finished its processing. The task is thus assured freedom from interruption from lower priority devices.

The PC is equipped with a single 8259 device that provides eight levels of interrupts; the AT has two that, combined, provide 15 levels. The general 8086-family of processors can be configured with as many as eight interrupt control (8259A) devices slaved to a single, master controller. Each slave provides eight independently controllable interrupt priority levels—for a maximum of 64. In interrupt processing, RTX creates a direct relationship between interrupt priority level, interrupt vector location, and interrupt task priority (refer to table 2).

A task installs an entry in the system's interrupt vector by making the iRMX call SET$INTERRUPT. This call can install a simple ISR, or, optionally, establish the calling task as the installed ISR's companion interrupt task.

If it establishes itself as an interrupt task, a task inherits special properties and gets special treatment from iRMX. It is immediately assigned an execution priority that corresponds directly to the interrupt priority level of its device. It also becomes more restricted in its activities than would a general-purpose task. Once it has made itself an interrupt task by calling SET$INTERRUPT, a task must confine itself strictly to the business of processing interrupts: waiting for a signal from its companion ISR, processing the interrupt, and looping to await another interrupt. A task remains under these restrictions until it calls RESET$INTERRUPT, which dismantles a device's interrupt processing apparatus by disabling its interrupt level, removing its ISR from the interrupt vector, and deleting the calling task.

ISRs with companion interrupt tasks communicate with them using the two system calls. The task calls WAIT$INTERRUPT to await a signal from the ISR, which calls SIGNAL$INTERRUPT to activate the task. In a busy environment, an ISR may be activated and call SIGNAL$INTERRUPT many times before its companion task gets a chance to run. When it first establishes itself via a SET$INTERRUPT call, an interrupt task

specifies the maximum number of unprocessed SIGNAL$INTERRUPT calls that iRMX should allow to accumulate before disabling the task's associated interrupt level. If a device gets too far ahead of its task, iRMX applies this automatic throttling to the device by disabling its interrupts until the interrupt task has run and called WAIT$INTERRUPT, and thus has reduced the count of outstanding interrupt service requests.

ISRs and interrupt tasks can communicate data and control information through a shared data segment. ISRs with companion tasks have the option of causing the iRMX system to call ENTER$INTERRUPT when they are activated. ENTER$INTERRUPT loads a data segment that is shared with the interrupt task (see figure 2). Whether or not they share data with an interrupt task, ISRs must call EXIT$INTERRUPT before returning. EXIT$INTERRUPT sends an end-of-interrupt signal to interrupt control hardware and allows interrupts of equal and lower priority to resume.

ISRs can call only on a highly restricted set of system calls. iRMX thus enforces its guideline that interrupts requiring substantial processing be serviced by an interrupt task instead of by a simple ISR. An ISR cannot, for example, refer to semaphore or mailbox objects; all interaction between an ISR and system tasks must be achieved indirectly by the ISR's companion task. This is a departure from the common approach to ISR/task communication—most multitasking systems permit ISRs to make (nonsuspending) system calls, such as posting units to a semaphore, in order to synchronize with tasks.

EXTENSIONS TO iRMX

A systems architect willing to undertake operating system-level implementation can expand iRMX in two important dimensions. New system calls—*operating system extensions*—can be implemented and new object types—*extension objects*—can be added to the iRMX repertoire. Operating system extensions can be called by application tasks through the interrupt vector. iRMX reserves the last 32 interrupt vector locations for use in calling operating system extensions (the extension mechanism allows the addition of a virtually unlimited number of system calls, by multiplexing the use of the 32 vectors).

Operating system extensions. To create an operating system extension, a systems builder must provide two components: an *interface procedure* and a *function procedure*. Interface procedures are linked with application jobs; they collect the system call's parameters in a suitable form and execute the extension's assigned software interrupt. The function procedure sits on the "other side" of the interrupt; it receives control from the interface procedure and performs the work that is necessary to the function of the system call.

Because they are accessible via the interrupt vector, it is not necessary to combine function procedures, using the object module linker, with the application jobs that use them. Function procedures run with privileges not available to application code. They can, for example, make objects immune to deletion using the DISABLE$DELETION call.

However, they inherit additional responsibilities along with the privileges. Extensions should be constructed to participate in iRMX's exception reporting and handling scheme just as standard system calls do. Intel recommends that, before invoking a system call's function procedure, an extension's interface procedure save the calling task's exception handling address and mode and substitute its own. By interposing its own exception management, an extension can appropriately hide, filter, modify, or pass along exceptions that its function procedure provokes in the course of processing an application task's request. The interface procedure is responsible for restoring the calling task's exception handling apparatus before returning control to the task.

Extension objects. With system calls CREATE$EXTENSION and CREATE$-COMPOSITE, iRMX supports the creation of new object types in addition to the Nucleus's eight built-in types. CREATE$EXTENSION returns a token that is, in effect, a license to create objects of a new type. CREATE$-COMPOSITE actually creates new objects, building them as compositions of other objects. By building composites of composites, objects of arbitrary complexity can be constructed.

To support a new object type, a system must provide a *type manager module*, consisting of an initialization part and a service part. The initialization part creates the new object type and establishes a mailbox to which the new objects can be sent for deletion when their owning jobs are deleted. The service part, which may or may not contain operating system extensions, provides the primitive functions by which application tasks are able to create and manipulate the new objects.

Several reasons might influence a designer choosing to implement new iRMX objects instead of simply building

an application to create and manage, without the operating system's cooperation, its own functional modules or data structures. Extension objects can simplify and standardize the view of systems-level function that is presented to application programmers. The same rules that application builders have learned to observe for passing standard iRMX objects through mailboxes, for example, apply to extension objects.

Extension objects simplify the job of the systems designer as well, by providing well-defined interfaces for creation, deletion, and communication of objects among tasks. A designer implementing a "ring buffer" object for use in an application, for example, need not start from scratch in designing an application interface for the new structures. Instead, he can model the set of calls for manipulating ring buffers after iRMX's calls for manipulating mailboxes.

I/O SUPPORT SUBSYSTEMS

The RTCS adaptation of iRMX for the AT is configured with both iRMX optional I/O support layers present. Application programs use the services of the BIOS and EIOS by making system calls. Although the extended system logically sits on top of the basic system, applications can call directly to either layer.

Together these two support layers provide a full complement of function, including buffered, device-independent I/O, and management of hierarchical systems of named directories and files. Applications requiring flexible, high-performance, no-frills I/O service can call directly on the BIOS to perform asynchronous, unbuffered I/O. Those that can afford more processing overhead can take advantage of the less flexible, but easier to use EIOS.

Device drivers in an iRMX system are considered logical parts of the BIOS. RTX includes drivers for the system console (keyboard and display device), floppy- and hard-disk drives, printer, and serial I/O ports. In current releases, applications interface to the PC's graphics adapters is not provided.

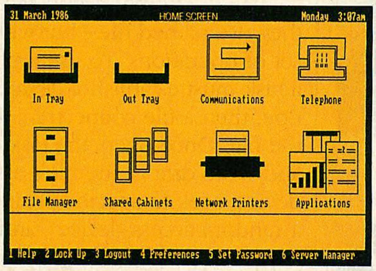
The BIOS device drivers form the bottom layer of an interface that, from an application task's viewpoint, provides device-independent access to system devices. Before using an I/O device, a task must call on the I/O systems to establish two *connections*. First, a *device connection* is established by RTX at system initialization; it attaches a logical name and a logical device object to a particular device's driver. Because it must name a physical device, a task establishing a device connection (that is,

At Last A Network That's Easy To Use

FIVE TERRIFIC REASONS TO TRY TAPESTRY® NOW

No matter what local area network hardware you choose – your next choice should be Tapestry networking software from Torus Systems. Here's why.

1 **IBM chose Tapestry** for its overseas market, and it quickly became the hottest LAN system in Europe.



This is the friendly screen that caught Big Blue's eye.

2 **It's the world's easiest-to-use networking software** – much simpler than Novell, 3Com, or IBM because Tapestry is managed with icons. To use it, you just point to graphic symbols: the "in-tray" to get mail, the "cabinet" to access files, and so on.

You even use icons to configure the system, so almost anyone can set up a Tapestry network – and keep it running smoothly.

3 **Tapestry is the most complete LAN software money can buy.**

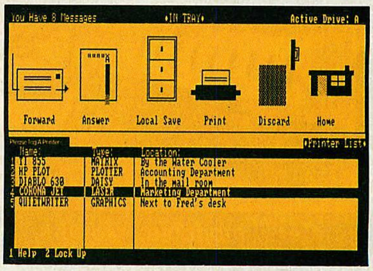
The basic system gives you:

· Advanced electronic mail
Compose memos with the Torus text editor. Send them to any person or group on the network, just by pointing to names. Track the mail you've sent. Get messages instantly.

· Cost-effective sharing of hard disks, printers and modems
You can have up to 100 Tapestry workstation/servers and access any one of them by touching the right icon. Since all Tapestry servers are undedicated, each station is free for business as usual. You don't have to invest in specialized hardware, or sacrifice your valuable PC's.

· Advanced communications capabilities
Need information from the company mainframe or an outside service? Just point to the desired Service icon. Torus offers a family of network gateways* that automatically make the connection and speed your information through.

· And there's more!
Tapestry also provides central storage of all your applications so they too can be accessed with icons. Automatic file locking so you can safely run single-user applications not originally designed for networks. File Manager icons that let you manipulate files without using DOS commands. And a Telephone Manager that places your calls and maintains a personal electronic rolodex.



What good is a network if your people don't know how to use it? Tapestry's screens are so easy to understand, anyone can net great results without special skills or training.

4 **Tapestry supports all the standards,** like IBM Token-Ring, PC Network and 3Com hardware. Like PC-DOS 3.1 and NETBIOS. So you can choose the LAN hardware that's best for you, and you won't have to worry if your software will run.**

5 **This coupon gives you great savings and a money back guarantee!**

Usually, Tapestry costs $495 for your first workstation, and $295 for each additional station. BUT NOW YOU CAN GET A TAPESTRY STARTER PACK (SOFTWARE FOR TWO STATIONS) FOR JUST $295!

If you aren't satisfied with Tapestry for any reason, you can return it within 30 days of shipment, and we'll gladly refund your purchase price (less $50.00 restocking fee.)

So send this coupon to us, take it to your Torus Dealer, or call in your order today, and start networking the easy way.

Co Name ____________________

Address ____________________

City ________ State ____ Zip ____

Phone ____________________

Networking Hardware ____________________

Number of Workstations in Office ________

Subtotal $295!

Add $7.00 shipping/handling $______

California residents add 6½% sales tax ($19.18) $______

Amount enclosed $______

Payment VISA MC Check

Credit Card Exp. Date ____/____

Card # ____________________

This offer is good only in U.S. and Canada, and may be discontinued without notice.

TORUS 495 Seaport Court, Suite 105
Redwood City, CA 94063
(415) 363-2418

System Requirements: Station 1: IBM PC, XT, AT or compatible 384K, 10MB hard disk. Station 2: IBM PC, XT, AT or compatible 320K. Plus suitable networking hardware.

Notes: *TTY is standard, while VT100, 3270 SNA, and Remote Access are purchased separately. **Tested applications include: dBase III Plus, MultiMate Advantage, Open Systems Accounting, Displaywrite 3, WordPerfect, MS Word, Lotus 1-2-3, AutoCAD.

CIRCLE NO. 184 ON READER SERVICE CARD

creating a data conduit to a particular device) is not deviceindependent. *File connections* are similar to wires that pass through device connections; they are established by calling either S$ATTACH$FILE or S$CREATE$FILE, depending on whether or not the file already exists. File connections identify devices by referring to previously established device connections, not by naming the device, and thus are established in a device-independent way. Application tasks routinely make and break file connections dynamically. Once a file connection is in place, a task calls S$OPEN to open the file for reading and writing in the conventional way.

The I/O system provides the management of three kinds of files: named, physical, and stream. *Named files* can be maintained on random access secondary storage devices, such as disk drives, in conventional directory structures. A *physical file* occupies an entire physical device, allowing it to be managed at a low level, as if it were a single string of bytes. *Stream files* provide a mechanism for intertask communications, in effect allowing tasks to take the role of devices (these are files with no coupling to physical devices). Stream files may be concurrently written by one task and read by another.

To use named files, a task must belong to a job that has established user information for iRMX inspection. Using the BIOS's named file management services, secondary storage devices may be subdivided into *volumes*, each of which contains a single tree of directories and files. A volume containing named files is managed like a DOS file system, but with the additional capability of individually controlling each file's accessibility on a user-by-user basis. Delete, list, add entry, and change entry rights to directories may be assigned to individual users. Files carry individually assignable delete, read, append, and update privileges. Jobs that travel with user information are termed *I/O jobs*.

Physical iRMX files occupy entire devices. Sequential devices such as printers, plotters, and various display devices are treated most conveniently as physical files. Under some circumstances, it is also useful to treat random access disk or disk-like devices as physical files. In formatting a disk, for example, or in interpreting the contents of some foreign file system on a diskette, a disk is best treated as a stored sequence of bytes with no special internal structure; physical files present this view of the contents of a storage device.

Tasks use physical devices much as they use named files, but with a simpler set of operations. To use a physical file, a task must establish a device connection, then a file connection, then open the file. Once open, the task can read and write data to the file's associated device, seek to specific positions (if it is a random access device), and perform any other functions supported by the device's driver. Unlike named files, physical files do not carry access control information. They can be manipulated freely by any task that establishes and opens a file connection.

Stream files, though accessed using the I/O systems, are not attached to any physical device. They provide a means for connecting two tasks in a reader/writer relationship. As part of system initialization, the I/O system establishes a device connection to a pseudo-device representing a data stream. Tasks wishing to engage in stream I/O must establish a file connection through this special device connection. Once a stream file connection is established, a task uses it to read or write data (depending on its role in the conversa-

WHEN YOU NEED ACCESS TO FULL MEMORY, STRUCTURE, COMPATIBILITY WITH GW- & PC-BASICA, OR WHEN YOU NEED THE POWER AND FLEXIBILITY OF C OR PASCAL...

. . .You need BetterBASIC. The BetterBASIC compiler has become the standard by which other BASICs are evaluated. BetterBASIC is completely compatible with GW-BASIC and PC-BASICA when running on IBM PC's and true clones. You can load and run your existing BASIC programs in BetterBASIC. It uses standard MicroSoft syntax and gives you more than 150 additional statements such as XREF, DEFINE WINDOW, MAKE MODULE, and PROCEDURE. In benchmark comparisons, BetterBASIC is five times faster than interpreted BASIC. There is optional 8087/80287 math chip support, and an optional Runtime System to create stand-alone EXE. files. BetterBASIC is not copy protected. Technical support is provided for all registered users.

See for yourself why Dick Aarons of *PC Magazine* said *"BetterBASIC may be the best of all BASIC programming worlds"* and selected BetterBASIC as "Editor's Choice" (Oct. 29, 1985).

BetterBASIC	$199
8087/80287 Math Chip Support	$99
Runtime System	$250
Sample Disk with Tutorial	$10

Ask your dealer or call to order: **1-800-225-5800**

In Canada, call 416-469-5244

Better BASIC

BetterBASIC is a registered trademark of Summit Software Technology, Inc. IBM PC, XT, AT are registered trademarks of International Business Machines Corp. Microsoft is a registered trademark of Microsoft Corp. Tandy is a registered trademark of Tandy Corp.

CIRCLE NO. 195 ON READER SERVICE CARD

Summit Software Technology, Inc.™
106 Access Rd. Norwood, MA 02062

tion); data are delivered from the writer to the reader via BIOS facilities.

In addition to serving a useful role within a production system the stream I/O facility creates practical flexibility for systems developers by allowing tasks to serve in the role of input or output devices. A device for which hardware or software is not yet complete can be simulated by a task in order to expedite system integration and testing. Alternatively, systems developers can take advantage of the I/O system's device independence to capture a stream of data in a named disk file for analysis, instead of passing it along to its ultimate destination.

The EIOS allows the specification of a buffering level in association with each file at the time a file connection is established. If zero buffers are specified, every task call for I/O through the connection results in a physical operation on the file's associated device. If one or more buffers are made available, the system automatically blocks data and defers physical read (write) operations until buffers are (empty) full. Multiple buffers allow the I/O system to perform read ahead/write behind-style management of file I/O.

Tasks using EIOS calls to read and write files automatically run synchronously with the file system; that is, they receive control back from an I/O call only when the requested operation is finished. Tasks also have the option of calling around the EIOS and using the facilities of the underlying BIOS directly. Using the basic system's calls, tasks can initiate a read or write operation, continue with other processing in parallel with the performance of the I/O, then call the system back to await completion of the operation and to check its status.

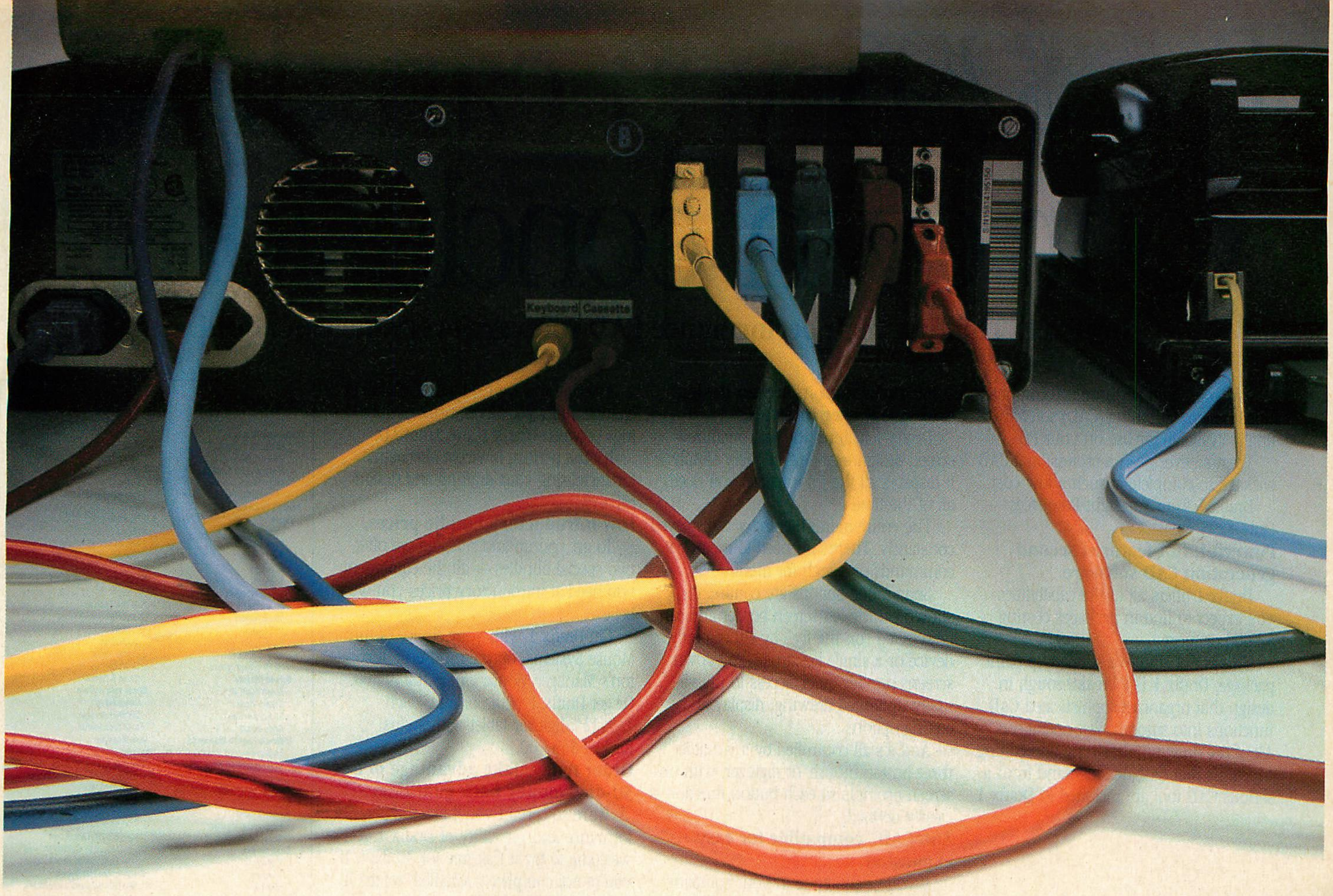
DEVELOPING RTX APPLICATIONS

Because RTX includes UDI, any of Intel's systems development tools built to access iRMX services will run under the RTCS product. These tools include an assembler, a linker and locator, and compilers for the FORTRAN, Pascal, C, and PL/M languages. The standard RTX package from RTCS includes Intel's assembler, linker, and locator; programming languages are available separately, also through RTCS.

In addition to hosting Intel's software development tools, RTX serves as a general-purpose software development environment by making other iRMX facilities available to the system developer. Developers using RTX as a base can communicate with iRMX using its Human Interface capabilities for interactively accepting and executing user commands. File storage is provided by the iRMX I/O systems, which maintain hierarchical, DOS-like systems of named files. On PCs configured with 512KB of memory, the RTX operating system can support multiuser operation, offering systems developers an option not available with DOS-based development.

RTX users have the option of developing software under DOS. RTCS Corporation's RTCS/UDI package equips a PC or PC/AT running PC-DOS with a fully Intel-compatible UDI, enabling it to run any of Intel's standard utility and software development programs. The PC UDI requires 256KB of memory, one 360KB floppy-disk drive, and PC-DOS version 2.0 or later.

To bring the UDI up under DOS, the user runs the RTCS program UDI, which installs a layer of resident intelligence in the DOS environment. Once UDI is installed, it sits between the user and the DOS command processor and allows the execution of three types of commands: ordinary DOS commands



Where's The Logical Connection™?

No more spaghetti!

At last there's an easy way to connect printers, computers, modems and other devices together — in *any* configuration! The Logical Connection™ gives you 8 programmable ports (4 serial, 2 parallel in and 2 parallel out) that you can connect any way you want. If you need more, you can "daisy-chain" them together to interconnect up to 270 separate devices. And if you have computers in more than one location, The Logical Connection™ can link everything in one office to everything in another *over a single twisted-pair wire!* (Or a single phone line with a modem at each end).

Share your printers.

Now you can give every computer in your office access to every printer — even if they're strung out all over the building. And because The Logical Connection™ gives you software control over which output goes to which input, you'll never have to unplug another cable to share your resources.

Fully programmable.

The Logical Connection™ provides automatic parallel to serial and serial to parallel conversion. Just plug in your devices and call up The Logical Connection's™ user-friendly configuration menu on your monitor. It will guide you step by step through the process of defining inputs and outputs, with extensive online help and easy to understand prompts. And you can change configurations just as easily, as often as you want.

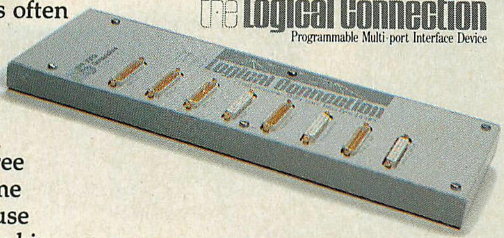
A big, smart buffer.

You won't have any slowdowns (or disasters) if three computers want to use the same printer at the same time, because The Logical Connection™ has a big 256K buffer built-in, with sockets for easy installation of another 256K. Smart buffer management spools each document for printing in the order received, so your computers can keep right on working.

The logical conclusion.

So if you're looking for a simple solution to the problem of connecting (and sharing) multiple printers, plotters, modems and other devices, it's time you got the connection. Get The Logical Connection™ at your leading computer dealer, or order direct from Fifth Generation Systems.

the Logical Connection
Programmable Multi-port Interface Device



Fifth Generation SYSTEMS
Innovative Products Using Today's Technology

CIRCLE NO. 125 ON READER SERVICE CARD

7942 Picardy Avenue, Suite B350, Baton Rouge, Louisiana 70809 • (213) 493-4483

For $295 you can draw your own conclusions.

Drafix™ 1. The first high performance CAD software everyone can afford.

Low-cost, high performance computer-aided design & drafting (CAD) software is finally available for use on your PC.

There's never been a professional CAD package so complete at such a low cost.

Drafix 1 from Foresight Resources. Only $295.

Powerful, full-featured second generation CAD software.

Drafix 1 offers all of the capabilities you expect to find in packages costing $2,000 or more.

This is not simply a souped-up paint package. Drafix 1 is a breakthrough in design that organizes sophisticated CAD functions into smooth, fast operations. All of the drawing, designing and editing functions that designers demand from a serious CAD tool are included in Drafix 1.

A breakthrough in screen design and visual user interface.

Drafix 1 features the best organized screen design ever devised. Very simply, it displays all the information you need, all of the time.

The entire menu hierarchy is displayed constantly. There's no need to memorize commands or search for menus.

Snap-mode options are continually shown on the left screen border and can be selected "on the fly" by either pointer device or a single keystroke. Roll-down screens provide quick access to the virtually unlimited drawing, display and editing options.

And it's all controlled by a versatile three-button mouse, or digitizer, with on-screen prompts so each button function is clearly defined.

AutoCAD® compatible for easy expansion.

If you or someone in your company already uses AutoCAD, Drafix 1 offers an inexpensive way to add to your capability.

The optional Drafix 1/AutoCAD file exchange utility permits transfer of drawings between Drafix 1 and AutoCAD.

It's the perfect, low-cost alternative for increasing your drafting and design capacity.

Low introductory bundle prices.

To get you up and running we offer two special bundles—all the equipment you need at introductory prices you won't believe.

Get Drafix 1 with your choice of the Mouse Systems' optical mouse or Torrington's Manager Mouse® for just $395.00. Or get Drafix 1 with SummaSketch 12"x12" digitizer tablet with stylus for just $585.00.

Order today. Take advantage of our 30-day, money back guarantee.

We're so certain you'll like the performance and versatility of Drafix 1 you're welcome to try it risk free for 30 days. If you're not completely satisfied, return it to us for a full refund.

Find out just how good CAD software can be. For only $295.00.

Use our toll-free number and any major credit card to order your copy of Drafix 1 today.

SPECIFICATIONS

Code specifications
- Lattice "C" language
- Halo graphics
- Floating point database
- Coprocesser support
- PC-DOS/MS-DOS 2.1 or later

Drafix 1 Features

Items
- Lines
- Pt markers
- Arcs & circles
- Note text
- Polygons & ellipses
- Symbols/individual & nested

Item attributes
- 16 pen colors
- 255 Layers
- 8 Linetypes
- 12 Text fonts
- 32 Pt marker types

Screen Display
- Zoom/Pan/Full
- 8 Save Views
- Slide save/view
- Grids on/off

Metric & English Standards
- Engineering (decimal/fractional)
- Architectural (ft in)

Project drawing info.

Numeric input
- Keyboard and cursor
- Absolute, Relative, Polar

Snap Modes
- Gridpoint
- Endpoint
- Midpoint
- Intersect
- On item
- Quadrant
- Tangent
- Arc center

Transform/Copy
- Move
- Rotate
- Scale
- Mirror
- Align

Modify/Edit
- Attributes
- Break/Divide
- Fillet/chamfer
- Trim
- Stretch
- Erase
- Explode symbols & polygons
- Mask
- Replace
- Merge
- Region select
- Workgroup

Auto hatching, polygon fill

Auto Dimensions
- Linear Hor/Ver/Aligned
- Angular
- Dia/Radius
- Leader Notes
- Ordinate
- Chain & Baseline

Check calculations
- Coordinates
- Distance & angle
- Area & perimeter

Item Masking

Hardware

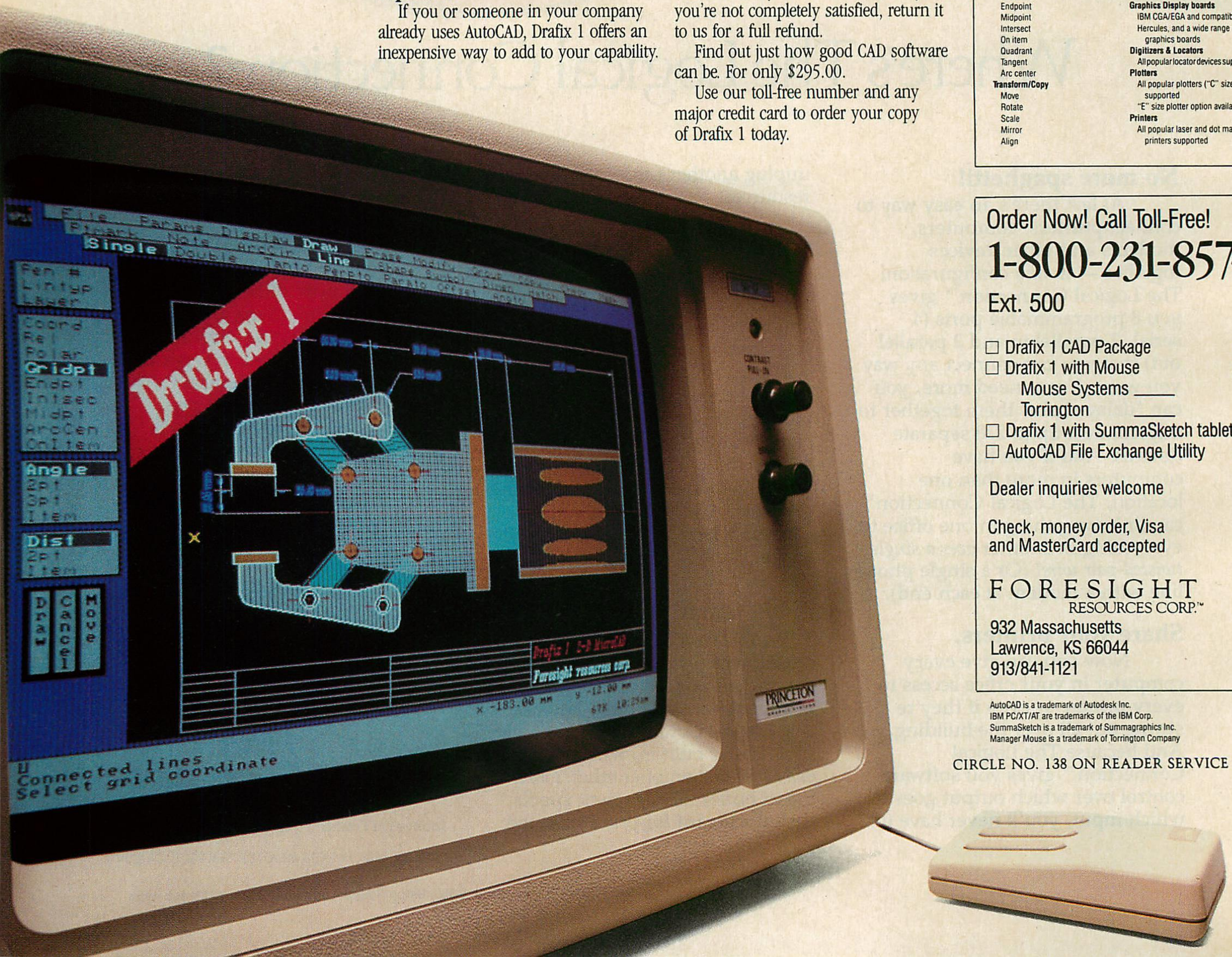
Computers
- IBM PC XT AT and compatibles
- AT&T 6300 Plus
- Zenith
- System Requirements
  - — 512K RAM
  - — RS-232 Comm port
  - — Mouse or digitizer
  - — Coprocessor recommended

Graphics Display boards
- IBM CGA/EGA and compatibles
- Hercules, and a wide range of graphics boards

Digitizers & Locators
- All popular locator devices supported

Plotters
- All popular plotters ("C" size) supported
- "E" size plotter option available

Printers
- All popular laser and dot matrix printers supported



Order Now! Call Toll-Free!

1-800-231-8574,

Ext. 500

☐ Drafix 1 CAD Package	$295
☐ Drafix 1 with Mouse	$395
Mouse Systems ____	
Torrington ____	
☐ Drafix 1 with SummaSketch tablet	$585
☐ AutoCAD File Exchange Utility	$ 95

Dealer inquiries welcome

Check, money order, Visa and MasterCard accepted

FORESIGHT RESOURCES CORP.™
932 Massachusetts
Lawrence, KS 66044
913/841-1121

AutoCAD is a trademark of Autodesk Inc.
IBM PC/XT/AT are trademarks of the IBM Corp.
SummaSketch is a trademark of Summagraphics Inc.
Manager Mouse is a trademark of Torrington Company

CIRCLE NO. 138 ON READER SERVICE CARD

and programs, UDI built-in commands, and Intel software tools and other programs that make UDI system calls. The UDI program also can be used on a single-shot basis, to run an Intel program one time without installing itself and remaining resident.

In addition to the DOS-executable version of the UDI, RTCS/UDI includes a subroutine library (in IBM DOS linker format) that contains the PC-DOS UDI (as listed in table 1). Using this library, users can construct programs that, like Intel's software tools, can be executed in any 8086-family operating environment supporting the UDI.

Although similar in structure, RTX (iRMX) file systems and DOS file systems are not compatible. They can, however, coexist in separate partitions on an XT or AT hard disk. RTCS provides two file translation programs, PC2RTX and RTX2PC, with which files can be transferred back and forth between the two systems.

PIECEWISE PROGRAM LOADING

In conjunction with its multiuser, general-purpose capabilities and full-featured file system, RTX offers more flexibility than most realtime systems do with respect to the way executable systems are prepared and loaded into memory for execution. An entire application system does not need to be loaded and executed as a monolithic unit. In addition to an initial bootstrap loader, RTX includes Intel's Application Loader, an iRMX subsystem that provides a set of system calls that can be used to load programs from secondary storage into memory under control of application tasks. The Application Loader can load program overlays as well as entire programs from secondary storage, and is capable of dynamically allocating memory and bringing programs into an already running system.

Using Intel's LINK86 linker and LOC86 locator, object routines can be prepared in compact, medium, or large segmentation models. They can be generated in absolute, load-time-locatable, or, in the case of modules in the compact model, position-independent form.

The Application Loader offers two levels of function for bringing prepared executable code into memory. The simpler level, which is implemented in the A$LOAD system call, reads and locates code in memory, leaving it up to the calling application task to determine how and when it is to be executed. The more complex level, implemented in the A$LOAD$IO$JOB and in the S$LOADIOJOB system calls, loads the code, creates an I/O job, and starts the job's initial task running.

The loader can load I/O jobs asynchronously or synchronously. With asynchronous loading, the invoking task can continue executing in parallel with the loader, and must specify a mailbox through which it will later receive the loader's report on the operation's status in a *loader result segment*. Asynchronous operation is not available for loading overlays; a task loads an overlay by making the S$OVERLAY system call, and must wait synchronously for the completion of the load.

REALTIME ADVANTAGES

By any measure, iRMX is a big operating system. Its basic documentation stands two feet high. It can consume 200KB of memory in maximum configurations (the standard AT/RTX configuration occupies about 174KB). iRMX is also complex—learning its object-oriented world view and navigating its myriad configuration and operating options requires close study and careful attention to detail. It can impose considerable execution time overhead on application tasks. By comparison with minimal systems, iRMX is harder to learn, harder to fit, and slower to run.

Its counterbalancing advantages are many. iRMX provides an array of function that is vast in comparison to most microcomputer operating systems. Its multiuser file management capabilities, for example, far outdistance those of DOS. Microcomputer implementations of UNIX and its derivatives provide similar levels of function, but lack iRMX's realtime facility. iRMX's flexible and thorough-going interrupt management capabilities, though they impose execution time overhead, are unmatched for function in the system's market. With its job-oriented resource management, directory and file access controls, and Human Interface services iRMX stands alone in the realtime market in its support for multiuser, time-sharing systems.

iRMX's object-oriented architecture, though it presents a larger training hurdle than that of more traditionally crafted operating systems, holds many advantages for organizations adopting it as an implementation base. It encourages and even enforces an approach to system design that embodies some of the more significant advances in software design methods of the past decade. "Information-hiding" interfaces, for example, which isolate the implementation consequences of design decisions within single modules, and a generally high level of data and functional

abstraction are natural by-products of iRMX-based design.

Organizations adopting iRMX also will find engineers engaging in more thorough consideration of exceptional conditions—how and when they arise and how to handle them when they do. iRMX's exception-handling approach makes the design issues of error handling much less likely to be overlooked or treated as an afterthought.

Much of the operating system's complexity stems from the considerable work Intel's developers did to achieve two valuable goals: a high level of configurability and a powerful set of approaches to classic realtime design problems. Their efforts were successful in both respects. The sheer quantity of design attention in providing ways to deal with common realtime dilemmas is evident throughout the system. In intertask synchronization, for example, iRMX support goes well beyond the adequate functions provided by semaphores by offering the carefully designed region mechanism as well.

In addition to the benefits of iRMX's sound and comprehensive design, a number of practical benefits accrue to RTX users and resellers. iRMX's longevity, coupled with Intel's commitment to supporting the 8086 family, has made it a stable, mature product. The product works, it is in widespread use in the industry, and it carries the Intel name and reputation for quality. Its perceived reliability, a quality as important to marketeers as its actual reliability, is high. Like the PC and PC-DOS, iRMX's open, extensible architecture and widespread distribution has led to strong third-party support—many device manufacturers in the industrial instrumentation and control market offer iRMX drivers with their products.

FAVORABLE PRICING

In creating the PC/RTX and AT/RTX products, RTCS has made iRMX much more accessible, in both the technical and the business sense. Its packaging of a preconfigured version for the PC has eliminated a substantial technical barrier by greatly reducing the amount of work required for a reseller to bring iRMX up in support of a PC-based application. RTCS's more favorable low-quantity pricing removes practical barriers some resellers would encounter in Intel's pricing structure.

PC/RTX in standard configuration (quantity one) is $995. Usable as a runtime environment, development environment, or both, PC/RTX requires a PC or PC/XT with 256KB of memory, two 360KB floppy-disk drives—although 512KB and a hard disk are recommended—and a standard monochrome or color display. Included with PC/RTX is Intel's 8086 Family Utilities package, with ASM86, LINK86, and LOC86; RTCS's PC2RTX and RTX2PC file translation utilities; and an assortment of other Intel utilities. RTCS also offers discounts, from 5 percent for quantities of 5 to as much as 40 percent for quantities of 50. AT/RTX is priced at $1,495, and requires 512KB of memory and a hard disk.

Users purchasing the basic PC/RTX or AT/RTX product must develop applications under RTX unless they also purchase RTCS/UDI for an additional $500. RTCS/UDI permits single-user, PC-DOS-based development. For either development environment, Intel's Pascal, FORTRAN, and PL/M compilers are available through RTCS for $1,295 each; the (Intel/Mark Williams) C compiler is $2,795.

Users who must change iRMX configuration parameters or add and remove device drivers must purchase PC/RTX-C or AT/RTX-C. These are fully configurable versions of iRMX that contain the RTCS-prepared device driver libraries and configuration source files required to perform a full iRMX system generation. Priced at $2,000 for the PC version and $2,795 for the AT version, these packages require the purchaser to acquire an iRMX license directly from Intel. Intel's price for a basic license is $6,000. Resellers of systems built with configurable versions of RTX must pay Intel royalties as well as RTCS's per-copy charges; Intel's royalty charge for a single-copy sale is $300.

All RTCS products come with 90-day support entitling the purchaser to telephone technical assistance and automatic product updates during the period. The support period can be extended for a year at a cost of $160 for PC/RTX and $210 for AT/RTX. All RTCS users can use RTCS's dial-up bulletin board service, which RTCS uses to distribute product support information. The bulletin board supports uploading and downloading, in addition to a teleconferencing forum, in which users can locate each other and exchange software and information.

PC/RTX: $995; AT/RTX: $1,495
Real-Time Computer Science Corp.
1390 Flynn Road, Unit E
Camarillo, CA 93010
805/987-9781
CIRCLE 339 ON READER SERVICE CARD

Richard M. Foard is a software consultant who specializes in realtime systems design.

PC BRAND™: CAREFULLY CHOSEN PROGRAMMER TOOLS

30-DAY MONEY-BACK GUARANTEE

We refund the purchase price of any product returned within 30 days in entirely resalable condition. **You can even try out programs themselves** *if product code begins with E, T, or L through N — even if it means breaking the disk seal. Some developers do pose limits, so for products beginning with other letters, opening sealed disks constitutes acceptance. But you can at least review the manual. There's just nothing stopping your buying from PC Brand.*

BRIEF *Is Anything But.*
A Whopper of an Editor

With a name that belies its thoroughness, Brief™ has every feature you've ever contemplated for your editor-in-chief. Text, from keyboard or files, is housed in multiple buffers, and scrolled through one or more windows you open, close, resize. A text buffer may be called to different windows to view two areas at once. A change in one changes both. Text blocks may be marked for printing, writing to files, movement to scrap buffers for cut and paste into other buffers, or deletion, with as many "undo" levels as you want.

To find and fix, Brief has text search abilities rivaling "grep", with wildcards for matching, indifference to intervening characters, acceptance of character ranges, even multiple choice of patterns and their replacements.

If you use Lattice, C86™, or Wizard, and have 320k, you can compile your C program without ever leaving Brief. It finds the lines with errors, and marches you through the text for repairs.

Parts of Brief were written with its own Lisp-like macro language which has structure, 32-character variable names, conditional execution, loops, and you can actually read it! Nothing like the hieroglyphs we've seen elsewhere. Bulletin board and public domain disks with macros. Disk of contest-winning macros comes free with your purchase! "Simply the best text editor you can buy", Dvorak *Infoworld*. (Needs 192k.)

Ask for:	List:	PC Brand:
U0590	$195	$CALL

MULTI-HALO *Multi-Board*
Graphics Library

The premier graphics library that got the ball rolling for PC-based graphics and has grown so omnipotent that it supports over 25 graphics boards — including IBM's EGA and Nr. 9 Revolution's hi-res series — and has a multitude of mouse and printer drivers. All that in each box. Separate C versions for Lattice, M'soft, Aztec, CI86. What does Multi-Halo do? Just about anything. A full "paint" was written using it. Wonderful value for single license. Costly royalties though for redistribution. Specify: S0315 & Language. List: **$250.** We: **$199.**

WINDOWS for C/WINDOWS for DATA
Give Your Program a Clearer Outlook

Windows for C™ is a library of over 65 functions to add the pizazz and practicality of window partitioning to your application. Unlimited windows, each defined in a C structure for easy reference throughout your program, can be made either to pop up or permanently overwrite the screen. Routines will scroll and highlight lists with arrow keys, will read and scroll ASCII files vertically and horizontally in windows, and even write to memory-loaded files off the screen.

Logical treatment of video attributes permits unchanged programs to run on color or monochrome. Colors of windows are set individually.

All functions are in separate modules; only those used are linked. Only buffers holding on-screen or temporarily obscured windows occupy RAM; others released dynamically. TopView™ compatible. Best overall rating and fastest display in Bill Hunt's 7/85 *Tech Journal* review of five windowing products.

Windows for Data comprises all of Windows for C but takes in data through the windows as well. At the high level a single function lets you specify prompt string, field length, data type, screen location, picture, target variable, then sets lesser functions scurrying to get and process a user's input — any of which functions are available directly. There are utilities to get system date and time, mess with strings, create your own field masks.

Field options can require entry, prevent entry, permit insert or overtype, beeping on invalid or overflow keystrokes, and attachment of field-specific help messages and functions you want called to display messages or validate entries. And you decide which keys will clear a field, jump to the next or prior, quit, etc. Options diverse enough that a set of "fields" can be made to behave like a Lotus™ menu.

Many compilers. Free demo.

Specify Compiler:		List:	PC Brand:
T0100	Windows for C	$195	$169
T0150	Windows for Data	$295	$259

C-WORTHY LIBRARY — PIPE THIS ONE ABOARD!
Fits Out Applications with Shipshape Interface

Many libraries launch flotillas of functions for small crafting — re-working of strings, positioning the cursor, etc. C-Worthy, by contrast, is a formidable battle wagon for major C engagements.

The C-Worthy Library™ wraps an entire user interface around your application. Its full power can be summoned by only a few high level calls. Sound exaggerated? A single function call can set up a complete text editor in a screen window.

- High level calls pop menus and scrollable choice lists to the screen, restoring the background when dismissed, and branching to the chosen activity in your application. A full function set handles doubly-linked lists defined by C structures.
- Windowing facilities open portholes of up to screen size for viewing virtual screens larger than the physical screen.
- Full context-sensitive help screen management takes over this chore. Keyboard entry routines look for the help key on their own and interrupt with pageable text windows explaining what to do next.
- Full error message interface sends error codes and the functions which return them to C-Worthy which counsels user; you get to remove all that error-checking clutter from your core program.

Your application is nested in these powerful emissaries to the outside world. C-Worthy's imaginative architecture then makes heavy use of C's pointers to functions to find its way into your application to act upon the user's request.

Separate utilities maintain help and error message text and lists in files. This text segregation means applications can readily translate into foreign languages without reprogramming — doubly so because C-Worthy display routines automatically resize for text length.

Where the high level interface does not suit you, the low level routines are available as decomposed functions. All machine dependency such as key mapping is housed in interchangeable overlays loaded alongside the application at run-time; C-Worthy applications can thus run on a mix of PC and MS-DOS machines without recompilation.

C-Worthy hands you a consistent and intuitive interface and a revolutionary design approach. Novell found it "played a key role and accelerated development" in making its NetWare™ utilities easier for users. "You owe it to yourselves to take a look." Binary. Lattice. Others coming. Ingenious demo: call for it.

Ask for:		List:	PC Brand:
T0500		$295	$269
T0550	Novell Network	$495	$449

C-TREE
B-Tree File Manager, Source Code, No Royalties!

C-tree™ has been around since 1979. (It became Digital Research's Access Manager™). That means seasoned, sturdy code which hasn't cracked under the weight of prolonged and widespread use. C-tree comes in C source code, revealing all you've ever wanted to know about how b-trees are written. Provided you bind it into your binary application, you can redistribute C-tree without royalties.

C-tree's design splits nodes to allow any number of users to access an index file simultaneously even when updates are in progress. So multi-user configurations and adaptation to networks are possible. You must write record-locking routines, as they are compiler and operating system dependent, but shows how.

Thanks to source code which does not deviate from the K&R standard, C-tree can travel. Tests in many environments prove that C-tree gives your application a ticket to anywhere.

C-tree permits any number of keys for a data file, supports duplicate keys, variable record length files, multiple key indexes in a single file, etc., etc. — it's a comprehensive product with everything you'd expect. Intelligently designed, too. Both high level ISAM routines which minimize coding by handling all details of an activity; as well as decomposed step-by-step functions you can access directly. Either way C-tree maintains optimal index structures which will find the record you seek amongst a million ten-byte keys in no more than five disk seeks.

Ask for:	List:	PC Brand:
F0660	$395	$329

dBC
Lattice Library Maintains dBASE Compatible Files With the Power and Speed of C

dBC™ links C to dBASE. It creates and maintains files and their indexes which exactly replicate dBASE file design. So dBASE can read and update them. And the reverse. dBC can use any files created by dBASE. Now C and dBASE can operate on the same data bases interchangeably.

That opens up the widespread culture of dBASE installations to exploitation by C programmers. You can tap that market, avoid the resident dBASE language, and gain all the advantages of C with this single product.

dBC's functions parallel all dBASE's file handling commands, many decomposed to give closer control. The manual discusses each backed by demo source files on disk.

Use dBC for custom work for clients, or on its own. It's a complete ISAM file manager for C whether or not dBASE will be used in tandem, supports all four memory models, and can have sixteen index and data files open. Big discount to buyers of both dBASE II and III versions. Specify Lattice, Microsoft 3.x, or DeSmet.

Versions:		List:	PC Brand:
L00II	For dBASE II	$250	$195
LCCII	With Source	$500	$390
L0III	For dBASE III	$250	$195
LCIII	With Source	$500	$390

CURSES *Unix Style*
Screen Management

Curses from Lattice™ manages the screen of the PC like Unix™ curses. Library of 84 functions and macros parallels Unix with matching parameter lists. So your Unix program will feel at home when you move it to the PC, and programs created on the PC will be Unix compatible. Keeps any number of screen images in memory, full or partial size. Supports color, all four memory models. Vast function set to get characters, wrap lines, scroll, blank lines, highlight, etc. Carefully follows Unix curses terminal orientation by re-painting physical screen only on your refresh command. Ask for: L0850. List: **$125.** Here: **$99.** With Source: L0860, **$250 / $199.**

How Quick? *We stock everything you see here — and then some. Which means you can get it right away.*

Looking for Something? *We carry — or can get — many more products. Just ask.*

For Orders, Literature, or Catalogs, Call Us at…

800 PC-BRAND

That's (800) 722-7263. In NY State call (212) 242-3600
PC Brand, 150 5th Ave., New York, N.Y. 10011-4311
Telex: 667962 (SOFT COMM NYK)

Prices, terms, and specifications subject to change without notice.

SAVE $5 Just tell us this code J6

CIRCLE NO. 171 ON READER SERVICE CARD

TODAY's TOP QUALITY AIDS TO PROGRAMMING PRODUCTIVITY

PRE-C

Thorough "Lint"-like Analysis Now on the PC

Unix users have always had "lint" to thoroughly clean programs before they disappear into a compiler.

Pre-C™ looms larger than "lint". It finds problems your compiler won't. Problems that a debugger will have trouble figuring out. Even problems which will cause trouble with other compilers.

Pre-C finds all the syntactical tripwires that will blow out a compiler and much subtler problems: code which will never be accessed, casts with suspect conversions, variables declared as external but never used, functions never called, machine-dependent expressions which inhibit portability, obsolete usage.

But the big service of "lint" is this: Compilers work with one module at a time. They know nothing of other modules which only meet up at link time. Pre-C looks at *all segments of your program at once* and reports to you any inconsistencies of inter-module references like conflicting data type declarations; parameter lists in calls which disagree with the functions themselves in number or data type.

Pre-C uses the Unix System III compiler standard to guarantee maximum portability anywhere in the C world. Plentiful command line options relax such rigor during early coding when work is incomplete.

Use purchased binary libraries? Profile them for Pre-C and it will thereafter insure that function calls from any C program are letter perfect. A miracle worker in speeding large system development. Needs 128k minimum; 192k recommended.

Ask for:	List:	PC Brand:
P0590	$395	$295

DAN BRICKLIN'S DEMO PROGRAM

Slide Show Your Latest Greatest Idea INDISPENSABLE!

When the inventor of the electronic spreadsheet called with a new program, we sat straight up in our chair. VisiCalc™ was for businessmen, but Dan's latest is for us programmers.

Ever had trouble putting a program idea into words? Programs are screens! Words don't work. The answer? Show your program as a procession of screens.

Dan's new program makes it easy. It creates slide shows that imitate a program's screens and sequential activity. Create a screen — a snapshot of your planned product as it runs. Anything goes: words, borders, box rules, inverse and underlining of monochrome, foreground and background color on the CGA and EGA. Press a key and make a copy of this "slide", change it a little, by a single character perhaps, to show the next instant of run-time, then copy the slide forward again. Create a whole slide show of your program in action. It will seem like the program itself is running.

Each screen is in 80x25 character mode, not bit-mapped graphics. All 250 characters and attributes are available from scrollable lists which pop to the screen. All commands are layered in Lotus-style pop-up menus, with frequent choices mapped to the function keys as well.

Screen areas can be blocked for cut and paste or filled with color or characters, even made to blink. Slides of oft-appearing segments can be made for overlaying on other slides, and any slide may have several overlays assigned to it. Slides can be shuffled, deleted . . . many quick tools to save time, disk space.

Slides can proceed at time intervals or in response to keystrokes, and depending on a user's response, you can tell the program to branch anywhere in the slide sequence to create innumerable paths through your show.

Don't make your ideas struggle through coding to get to the screen. Dan's Demo is invaluable to prototype the program you are about to write, to position all the labels, choose the color decor, smooth out the keystroke interface *before* it gets etched in code. Or load the "capture" utility above the operating system, snapshot the screens of any running program, and load an instant slide show into Dan's program. Makes tutorials a snap.

Dan's Demo has blossomed throughout Lotus™, we hear. "Lotus [was] my major test site", says Bricklin.

Each purchase entitles you to redistribute fifty copies of the slide projector program along with your show. Plain manual, no binder, to keep the price low because Dan thinks everyone should have one (he's right!). You'll wish he had Demo then and left VisiCalc for now. (Needs 256k).

Ask for:	List:	PC Brand:
N0100	$75	$69

BASTOC OPTIMIZES!

Translates BASIC Programs Into C

For a trifling price, BASTOC™ will move truckloads of BASIC code over to C. It is a translator which takes in Microsoft Extended or CBASIC and emits pure Kernighan & Ritchie C for the Lattice compiler. It will optionally convert your program into a single monolithic C function or will decompose it into separate functions, one for each GOSUB label.

Version 2.0 adds optimization, with dramatic reductions in execution time. It converts to C integers those numeric variables it finds in BASIC programs which really do not need floating point. It eliminates unreachable code. Where BASIC uses full assignment statements to increment and decrement counter, BASTOC converts to C's compact form, nested in other statements. Strings are dynamically allocated in the target program, ridding your application of BASIC's catatonic halts for garbage collection.

BASTOC will try to create structure of even the most convoluted BASIC code, and writes any indigestible statement into the C output as a comment accompanied by an explanation of the problem. Also, you can optionally tell BASTOC to insert BASIC source lines into the C target as comments, a dandy way to learn the differences between the languages.

Specify:	List:	PC Brand:
S0375 & Which BASIC	$495	$399

GREENLEAF LIBRARIES

FUNCTIONS

New 3.0 has 225 functions in *both C and assembler source* as well as library format. We have versions for Lattice, Microsoft, C86, Mark Wms. New emphasis on tighter functional groupings to minimize excess baggage of functions loaded whether used or not. Manual's 250 pages now help select functions, as do demos and bulletin board. 32 DOS extensions: file and directory manipulation for 2.0 and 1.1. 23 Screen Functions: Select mode, page, monochrome or color, palette; cursor shape, positioning; clearing and scrolling; pixel get and put; read light pen. 60 String Functions: Manipulation of strings, including center and justify; efficient list operations which add, delete, and sort pointers to strings for top speed. 50 Graphic & Printing Functions: Primitives to access all graphics; typeface, formatting, and forms control. Plus keyboard status and function key assignment, time and date algorithms, read registers, memory size, peek and poke.

Ask for:	List:	PC Brand:
S0770	$185	$139

COMMUNICATIONS

Want your application to communicate with other users or remote date bases? Now you can build asynchronous communications right into your C programs! Over 60 functions and demo programs *in both C and assembler source code* set up an interrupt driven scheme with separate transmit and receive ring buffers (characters are simultaneously loaded at one end and transmitted from the other, or vice versa) for an arbitrary number of ports. Interrupt control means you can download a record, then halt the incoming stream to file it, display it, let the user tamper with it, send it back up line. It all happens within the context of *your* program. Goodbye separate communications software.

The Greenleaf Comm Library supports ASCII or binary, any parity, any word length, 8250 UARTs, all four Lattice C memory models, Hayes 300,1200,1200B and other modems.

Its 80-page manual has examples of each function and guides you through asynchronous communications.

Ask for:	List:	PC Brand:
S0750	$185	$139

PANEL

Feature-Laden Screen Design Tool

The newest version of this premier programming tool lets you layer your screen designs with up to ten overlapping images, making it easy to background a screen with pop-up lists, help boxes, and alternate sets of input fields.

Writing your own screenware is a good way to blow completion dates and profits. Panel™ works with you interactively to set up foolproof screen displays and data entry forms rapidly. It tests your form to prove that it (and test data) behave correctly, then converts the finished work into C source code for incorporation into your application. Compile with Lattice or Microsoft.

Wonderfully diverse attributes may be selected for any field — size, data type, color, of course, but also conversion of input to upper case; clearance of existing data when new entry is started; masks for standard formats (eg, dates, phone numbers); a choice of styles for numeric fields; phrases which fill in when their first letter is typed; multiple-choice lists from which to choose a field fill-in by cursoring a highlighted bar. Fields may be multi-lined (eg, name and address as one field) and scrolled if larger than the screen space allotted them.

Panel builds in a user interface for keystroke movement within and between fields, and supplies extensive validation routines for checking user field entries — in source code, so you can tack on your own unique variants. Screen designs may be dynamically loaded from file, or compiled into a program, and version 6 has optimized code to quicken display speed.

The whole package is wrapped in a monitor and keyboard customization package to tailor your application for other equipment. Panel. A superior productivity tool now bigger than ever.

Ask for:	List:	PC Brand:
S0400	$295	$229

BTRIEVE VERSION 4.0!

Queen-B File Manager Abdicates Royalties

This queen of b-tree file managers was unapproachable to programmers for whom royalties would ruin profit margins. So it's quite a ukase indeed that one need no longer pay a tithe to incorporate Btrieve™ in applications.

Now there is version 4.0, which hugely speeds DOS interaction, especially for large files with multiple keys. It also adds support for variable length records of virtually any length. Other new features: a read after write option to verify accuracy, useful in gritty environments like manufacturing floors; file password to deny unauthorized access or read only; and data encryption to assure network privacy.

Btrieve's foundation is a balanced-tree indexing scheme, conceded to be the fastest search technique devised (it will find any key in a million-plus item index in four or less accesses). Btrieve takes complete charge of all file creation, indexing, reading, writing, insertion, deletion, space recapture, and forward and backward searching based on full or partial keys. It builds commands right into the language you use in the form of functions you call to tell Btrieve what to do.

Btrieve has mainframe specifications! A single file may have up to 24 indexes. Segments of keys may be indexed. Each index can independently accept or block duplicate keys. Fixed record lengths can be up to 4090 characters; variable length records 64k; indexes 255 characters; files up to 4 billion bytes. It can even extend a file across two drives—even two hard disks!

Interfaces to C, Pascal, BASIC, and COBOL with single purchase; sample programs in all four languages.

The network version works with any network that supports the MS-DOS 3.1 file sharing function.

Code:	Version:	List Price:	PC Brand:
S0650		$245	$199
S0652	Network	$595	$529

PLINK86 & PLUS

Dynamic Cache Overlays Maximize Memory Use

Plink86™ long the overlord of overlay linkers, now has a Plus version. As a linker Plink can be used with any compiled language which delivers Microsoft/Intel format object files. It yields automatic symbol tables and more memory maps than DOS LINK, but its overlay power has won its reputation as a miracle worker. Plink86 shoe-horns large programs into small memory by binding into your compiled program an overlay manager which knows how to swap modules of large linked programs between disk and shared memory space. Plink86's straightforward overlay description language allows you to describe your overlay hierarchy in a structure permitting up to 4,095 overlays stacked 32 deep.

So if your program needs large chunks of memory, you no longer forego sales to folks who have less. But if you've assumed 128k, and they have 640k, Plink86-Plus™ knows to use extra memory as cache for overlays — at full speed compared to disk swapping. It also can automatically restore a displaced overlay to which a subsequently called overlay must return, and assign library modules to either a program's root segment or overlay areas.

Plink, the programmer's choice even when CP/M™ was the poobah of

Code:	Product:	List Price:	PC Brand:
S0500	Plink86	$395	$289
S0499	Plink86 Plus	$495	$359

PRICED TO SAVE YOU MONEY, SHIPPED FAST ANYWHERE.

NEW LOWER PRICES!

RUN/C PROFESSIONAL VERSION

Interpreter Now Accesses Binary Libraries

RUN/C was an innovator for converting inaccessible compiled C to an interpreted language as easy to grab hold of as BASIC. Great for learning, but a problem remained for pros An interpreter expects nothing but source code, and that put the vast resource of professional binary libraries off limits.

No longer: RUN/C Professional™ has the tools dynamically to load and unload multiple binary function libraries while in its interpreter. Your code can now reach for functions in the commercial C libraries like C-Food Smorgasbord™ - opposite— potentially any library compiled with Lattice's large model. How? The manual shows how to develop the interface to a library, using the Lattice compiler (a must!). How about your own archive of functions? No reason why not.

The RUN/C Interpreter

The interpreting engine lies at the heart of both the improved original RUN/C and the new Professional version. Its creators had the inspiration to make once formidable C behave on screen much like PC BASIC with a full-screen editor like WordStar®. Just create a program and RUN it. If it stumbles, LIST it, EDIT it, add lines, delete lines, RUN it again, fix it again. Use familiar commands like LOAD, MERGE, SAVE, FILES, even TRON and TRACE, and a free profiler.

RUN/C is ideal for rapid program development. Put up code at high speed, tinker and re-arrange, try out things devil-may-care, and let RUN/C find your typos and malaprops.

RUN/C has a treasury of functions built into the interpreter — over 100 paralleling the most used functions found in standard compiler libraries. So when and if the time comes to compile, your source code will find counterparts.

There are lots more features — system interrupts, a shell command to invoke any operating system command without leaving RUN/C, even the ability to load a preferred editor in parallel and switch back and forth.

RUN/C Standard Version

Straight RUN/C has all above but the Loadable Libraries™ docking module. It utilizes source code only, whether created by its own editor, or from any ASCII file, such as programs you've already written, or commercial libraries which supply source code.

It makes a splendid teacher. The manual has not just instructions how to use RUN/C, but its 500-plus pages provide a thorough-going demonstration of the C language itself. Every feature, of C or RUN/C, is accorded its own micro-chapter. Over 100 of these chapters are devoted to RUN/C's built-in functions, and every one lists a sample program showing how it is used. The programs are also on the disks. So as you read them in the manual, you can run them on the screen. (Needs 180k-256k recommended.)

RUN/C Professional

RUN/C Pro has every feature of RUN/C regular plus the binary library link-up and an extra level of debugging aids. They are ingeniously installed behind a built-in function, so you can call for debugging conditionally. The called function paints a menu of debugging tools to choose from, including immediate mode to display variables, single-step tracing, and changing of variables.

RUN/C Professional can tackle projects of any size. Use it as a creative front end to feed a continuous stream of source code into compiled modules. Only the source work in progress is still interpreted; the finished modules will whiz by at object speeds. It will change how you work. (320k minimum and 512k recommended to fit libraries.)

RUN/C: quite a run for your money.

Ask for:		List Price:	PC Brand:
S0910	RUN/C Classic	$120	$109
S0950	RUN/C Pro	$250	$185

OUTSIDE U.S.?

The dollar is weaker. In your currency, our prices are lower than ever. PC BRAND ships anywhere. We'll prepare the export documents and ship to you or your agent by air freight, courier, or air parcel post. Pay by credit card or wire funds (see "Terms" below).

LATTICE C VERSION 3.0

Major Upgrades to the Best Selling C Compiler

Lattice has labored and come forth with the long-awaited Version 3.0 of its top-rated compiler. A long list of enhancements, adoption of the ANSI draft standard, documentation rivaled by few, and add-on libraries matched by none in sheer quantity restore Lattice C™ to its leadership role as the C compiler to beat.

Lattice now embraces key UNIX™ enhancements which have entered the language since Kernighan & Ritchie: void functions returning no value, enumerated data types to assign stepped values to variables, the ability to pass data between structures by assignment statements. And 3.0 adopts checking of external function arguments by data type as proposed by ANSI to kill bug swarms when modules join up at link time.

The greatly expanded libraries, now comprising 325 functions(!), enable the file sharing and record locking provisions of DOS 3.1, provide a full complement of transcendentals, and a host of utilities to mimic the UNIX and XENIX™ environments.

Lattice now delivers smaller .EXE files, curing one past complaint, boasts very fast link times and a more efficient aliasing algorithm.

The compiler now defaults to the ANSI proposed standard when you need a strict mistress, but command line options tolerate straying. New options generate code to use 80186 and 80286 features, and the 8087 is of course sensed and utilized if aboard.

Lattice has enjoyed pre-eminence so long that developers have created far more tools to marry into Lattice C than any other compiler. Programmers now have an enormous resource of libraries and utilities to use with Lattice to speed their work. William Hunt, in his exhaustive analysis of 12 compilers in the 1/86 issue of the *PC Tech Journal* awards Lattice the only "very good" rating for add-on library availability. He sums up with this all-around accolade: "a fine product to consider for the production of important applications."

Ask for:	List:	PC Brand:
S0100	**$500**	**$CALL**

The GSS GRAPHICS SYSTEM

Leave the Driving to GSS

ANSI CGI STANDARD!

GSS™ has reconfigured two components of its comprehensive graphics tools to conform with the more advanced ANSI Computer Graphics Interface (CGI) standard.

At the heart of the system is now the Development Toolkit which contains all language interfaces and device drivers for keyboards, mice, joysticks, tablets, printers, plotters, cameras, and more. Drivers now house all management of vector graphics (plotters) and bitmaps used by raster input devices (scanners) to insulate completely the application program from concern for device idiosyncracy. No one else has implemented CGI that way. It means programming remains generic; just switch drivers and the same program will drive a different device, including intelligent controllers which do not want micro instructions.

GSS Kernel™ conforms to level 2b of ANSI's Graphical Kernel System (GKS) and contains all its needed drivers and language bindings. Kernel has macro level tools to draw and color an object, store the sequential instructions, and recreate the object on its own, as well as segment it, transform it, etc., all the while returning data on attribute settings, system and device status. So powerful, a single command may represent several score lower level statements.

Plotting has the equivalent GKS tools for graph and chart generation and their captioning: hand it apples and oranges, say "pie", and it bakes the numbers into a digestible display for screen or plotters.

Kernel and Plotting have tools to convert images they create to ANSI Computer Graphics Metafiles (CGMs), a tokenized standard for storing every form of graphic image as data. The Metafile Interpreter reads the contents of a CGM and interprets it with full CGI capability for re-creation on various devices, and you can cut and paste before display.

Quality software? IBM thinks so. They sell the GSS series under their own label.

Unit royalties and annual fees have been instituted for redistribution. Call for schedule. Needs 256k.

Ask for:		List Price:	PC Brand:
GS010	CGI Dvlpmt Toolkit	$395	$339
GS020	Kernel System	$495	$419
GS030	Plotting System	$295	$249
GS040	Metafile Interpreter	$250	$209

BETTER BASIC

New Version Compatible with Microsoft BASICs

This hearty implementation provides a real alternative to technical languages like C. It melds the most useful features of C, Pascal, and Modula 2 into BASIC, while retaining the familiarity of a language already known to millions. And now Version 2.0 is 100% compatible with Microsoft's GW™ BASIC and IBM BASICA including graphics, sound, and assembly language calls. Just load old programs and run. Save and they are converted to BetterBASIC.

It's big: BetterBASIC's hugely expanded features require 192k; your programs can go all the way to the PC's full 640k. It's comfy: Behaves like Microsoft BASIC at the interactive level, with a full-screen editor, direct statement execution, and always poised to RUN. It's fast: BB is an incremental compiler—unlike with interpreters each statement is checked and compiled just once. The Sieve benchmark runs *six times faster* than with Microsoft.

BetterBASIC® has C-like structures for reference to entire records so say goodbye to FIELD, MKI$, CVD, LSET, etc. It has "procedures" summoned by name unlike GOSUBs. Lots more features: built-in linker for compiled modules; trace; debugging breakpoints; cross-reference command; 32k strings; DOS and BIOS calls and interrupts; recursion.

Ask for:		List Price:	PC Brand:
S1200	BetterBASIC	$195	$169
S1201	Run-time Module	$250	$239
S1202	8087 Interface	$ 99	$ 89
S1205	Btrieve Interface	$ 99	$ 89

Need terms? *On-the-spot credit to most public companies, government, educational, medical institutions.*

Why Us? *A few thousand words later, you gotta admit, we go to great lengths to know programmer tools.*

TERMS AND CONDITIONS OF SALE

Licenses: Each price is for a license to use a product on a single computer and does not constitute its ownership. We will inquire for you about site licenses. Except as otherwise indicated or where "#" follows the Product Code, products may be used to create programs for distribution without royalty payments or additional licenses, provided said programs do not substantially replicate the products themselves.

Compatibility: PC BRAND's standard products are designed to operate with the IBM® PC, XT or AT under PC-DOS and require no more than 128k of RAM unless indicated. Non IBM machines using MS-DOS: contact manufacturer about precise differences so we can advise.

Returns: See box page one. Defective parts will be replaced. Please call for authorization to return a product for refund.

Payment: We honor MasterCard, Visa, American Express (no surcharge), checks in advance, or funds wired to PC Brand, c/o Chemical Bank, 126 East 86 St., New York, Account 034-016058. COD (U.S. only) for cash, money order, certified check (no fee). NY State, add sales tax. Purchase orders accepted from larger corporations and institutions at our discretion if you agree to net 30 days plus 2% a month late penalty thereafter.

Shipping & Handling: U.S.: UPS Surface: 1st product $6, each add'l $3. UPS 2nd Day Air: 1st product $10, each add'l $4.50. UPS Next Day Air or Federal Express 1-2 Day Air: 1st product $18, each add'l $6. FedEx Next Day 10 AM: 1st product $28, each add'l $7. International: Charges vary by destination and carrier. $10 per shipping container for export forms. Air parcel post at your risk beyond collected insurable amount.

For Orders, Literature, or Catalogs, Call Us at...

800 PC-BRAND

That's (800) 722-7263. In NY State call (212) 242-3600
PC Brand, 150 5th Ave., New York, N.Y. 10011-4311
Telex: 667962 (SOFT COMM NYK)

SAVE $5 Just tell us this code 16

Prices, terms, and specifications subject to change without notice.

CIRCLE NO. 172 ON READER SERVICE CARD

COMPANY MISSION

His company sent him on a mission to make two offices 1800 miles apart, work like one.

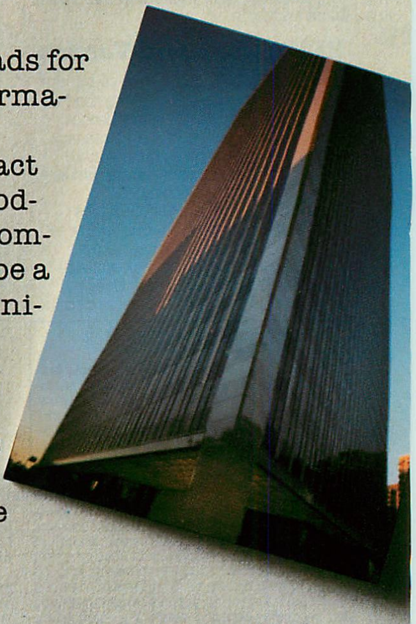
Every company has a PC systems expert like him—responsible for directing microcomputing strategies and budgets. His product recommendations are as good as purchase orders.

Where his mission begins

When he's on a tough assignment, he turns to PC Tech Journal first. Because it gives him the advanced technical information he needs, written by micro systems experts he can trust.

Inside two recent issues, he finds clues to his company's communication problem: Augie Hansen's review of the new communications package, "Relay"—and a preview of David Schwaderer's long awaited book, "Digital Communications Programming for the IBM PC."

Help him face the challenge



It's not only articles he reads for clues. He explores the information he finds in ads, too.

Ads in PC Tech Journal act as signposts to indicate products that can help him accomplish his task (whether it be a problem involving communications, custom boards, mass storage, or operating systems).

And PC systems experts find your ad without a lot of digging around. Because every product advertised is listed under its proper category in the product index—just one more reason why sophisticated

CASE 1 COMMUNICATIONS

systems experts turn to PC Tech Journal when they're ready to buy.

Accomplish your mission

Don't be the missing clue that can solve this PC expert's problem—and 100,000 other systems experts like him who read PC Tech Journal.

Call your PC Tech Journal sales representative, right now, and let Tech Journal guide customers to where you want them to go—right to your products.

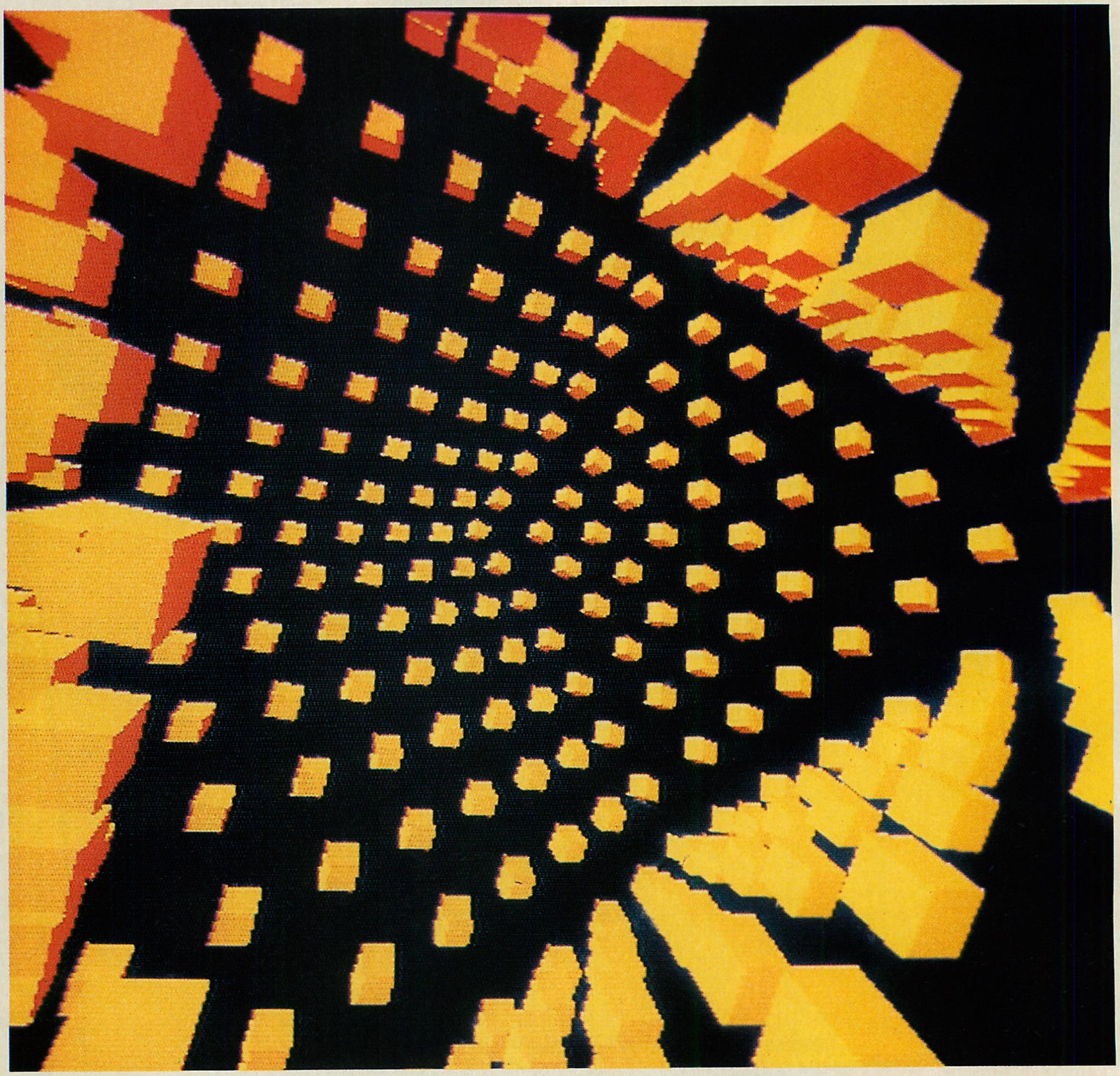
PC TECH JOURNAL

Unmatched power in your market

Ziff-Davis Publishing, One Park Avenue, New York, NY 10016 (212) 503-5100

DATA MANAGERS AS DEVELOPMENT TOOLS

A Data Manager for Flexible Designs



COMPUTER GRAPHIC • DOV JACOBSON

KMAN's richness of features makes for a powerful and flexible product, but one that can become overly complex without careful use.

RICHARD N. AARONS

Imagine a programming environment that is rich with functions and modern structures complete with a data manager that handles multifile accesses by itself, leaving the user to worry about program flow rather than the mechanics of file retrieval. Add to this a built-in spreadsheet or a powerful editor that handles program revisions interactively within the programming system itself. Top it off with the ability to build a communications facility or perhaps a business graphics module directly into an application. This is what KnowledgeMan/2 is all about.

KnowledgeMan/2 (usually shortened to KMAN) is a product of Micro Data Base Systems, Inc., the veteran designer of relational and postrelational database systems for mainframes, minis, and micros. MDBS III, the company's postrelational database and application development system, is considered a standard in the micro- and minicomputer industries. In many ways, KMAN is a subset of MDBS III features, and the two programs can interchange data easily through the optional K-Comm communications package.

The basic KMAN package includes a data management system (with query and programming languages) as well as a fully integrated spreadsheet. Optional utilities include K-Text, a full-screen text processor; K-Graph, a business graphics package; K-Paint an interactive forms generator; K-Report, a custom report generator; K-Comm, a communications module; and K-Mouse, a mouse driver/menu developer. Also available are K-Run and K-RunX, which are runtime systems for KMAN and the optional modules, respectively.

The fact that all the K-modules are sold separately suggests that the KMAN system is only loosely integrated; this is far from the case. The system actually comprises one .EXE file and 18 overlays. Most of the optional modules are made up of additional overlays.

Unique to this approach to integration is the Memory Overlay Management utility (MOMAN). It is used to add or remove the code contained in overlay files to the KMAN.EXE file, thereby alllowing the user to customize the system by selecting particular features. When KMAN is called at the DOS prompt, KMAN.EXE is loaded, and it, in turn, calls up overlays as needed to execute its commands. This works fine—especially for hard-disk systems in which overlay load time is insignificant.

If the user has plenty of memory available, the MOMAN utility can add the overlay file code directly to the KMAN.EXE file. Then the appropriate overlays are automatically loaded when KMAN is called from the DOS prompt. The user can elect to consolidate any number of overlay files into KMAN.EXE. This action increases the size of KMAN.EXE and, consequently, the time required to load it. Once KMAN.EXE is loaded, however, the overlays are always in memory and available within the KMAN environment instantaneously. Some KMAN users compromise by using MOMAN to add only those overlays that they use most often.

KMAN installation is relatively simple—especially in the recommended hard-disk environment. The program, its utilities, and demonstration files are provided on four diskettes. The optional modules are provided in separate packages, each containing one program diskette and appropriate documentation pages and tabs that can be added to the master documentation.

In the IBM PC, PC/XT, and PC/AT hard-disk environments, installation requires that all files be transferred from the distribution diskettes to a hard-disk subdirectory, that an appropriate driver be selected for the user's monitor type, that the KMAN subdirectory name be added to the PATH command in the AUTOEXEC.BAT file, and that the CONFIG.SYS file be updated to include the FCBS=255,0 command.

KMAN provides an installation utility that automates the task by asking where the user wants the KMAN files (subdirectories, etc.) and what type of monitor is being used. The installation routine updates the CONFIG.SYS and AUTOEXEC.BAT files, but it can confuse matters by adding its own PATH command to AUTOEXEC.BAT rather than appending its directory name to the existing PATH command.

KMAN's distribution diskettes are not copy protected, thus easing the installation process appreciably. MDBS also provides utilities that allow users to customize the standard keyboard or to set up the program for use with nonstandard consoles.

KMAN SYNERGY

The designers of KMAN talk of synergy when asked about the internal design of the data management system and the optional utilities. KMAN has been designed so that any command or feature, including the optional packages, can be invoked from anywhere in the KMAN environment. The system has no hierarchical structure—no "top level" to which the user must return in order to invoke commands. For example, a spreadsheet cell can contain a string, a variable (of any type), or an entire program. Data can be moved from data file

to spreadsheet to text processor to communications module.

KMAN supports four *classes* of variables, each of which supports four variable *types*. Variable classes—working, predefined, cell, and field—are an intrinsic part of the KMAN programming scheme. Variable types are numeric, string, integer, and logic.

All classes and types of KMAN variables are available for use in the interpretive environment. In fact, the variables, functions, and programming language elements can write programs that do not refer to data table files. For example, this is a valid KMAN program:

```
LET a = 12.0
LET b = 15
OUTPUT a + b
```

Programs either can be entered interactively or stored in text files called *perform* files. These are given the extension .IPF and can be run at any time within the KMAN environment simply by entering the command PERFORM or its synonym INCLUDE followed by the perform file name.

The number of working variables available at any given time is limited only by available memory. Working variable names must begin with a letter of the alphabet and can be up to eight characters long. In the KMAN environment, working variables are loosely typed; these variables derive their type through the assignment of initial values. The user should assign types to the variables at the beginning of a routine.

KMAN also allows dynamic variable typing—that is, a variable can take on the type of its current contents. Therefore, a variable's type can change over the course of a program. These matters of typing become important in screen forms when data are being presented on the screen or when screen input is being used to assign values to variables.

KMAN has been designed so that any command or feature can be invoked from anywhere within the KMAN environment.

The class of predefined variables can be divided into *environment* and *utility* variables. As their name implies, environment variables control the programming environment. The basic KMAN package has 60 such variables.

The environment variables control a variety of settings influencing I/O to and from the console, printer, and disk files; default mathematic precision and decimal positions; U.S. versus British date and decimal notation; default string length; foreground and background colors in the spreadsheet, text, and graphics modes; bell warnings; the availability of help menus; and other miscellaneous settings.

All environment variable names are in the form E.AAAA. The E-dot sequence identifies the variable as an environment variable, and the AAAA sequence describes its function. For example, setting E.OPRN = TRUE directs output to the printer, setting E.STAT = TRUE turns on automatic statistics generation, and setting E.BELL = TRUE sounds the bell upon invalid input.

Predefined utility variables, which begin with the character #, normally contain the results of some process, such as the latest statistics collected or information about the status of the current KMAN session. Many are updated automatically by KMAN. For example, the utility variable #FOUND contains the value TRUE if a search of a table found a record and FALSE if the record was not located. #DATE normally contains the system date, but the user may change the value, as in this example: LET #DATE = "01/01/33."

Cell variables contain the values of spreadsheet cells. The basic KMAN package includes an integrated spreadsheet holding a maximum of 255 rows and 255 columns. It can be used as a stand-alone spreadsheet or in combination with data tables or the optional text and communications packages. A subset of information contained in a data table can be converted to a specified set of cells in a spreadsheet. The information can be manipulated in the spreadsheet, then moved to the text processor (K-Text) for inclusion in a document. The document then can be distributed electronically using K-Comm. Of course, the whole process can be reversed.

Interestingly, KMAN treats cell variables as a two-dimensional array. For example, #(1,1) is cell #A1, while #(3,2) is cell #B3, and so on. The subscript of the cell array can be an expression as well as a constant, so programs can reference the spreadsheet just as any other array.

As noted earlier, the value of a spreadsheet cell can be a simple number or string, a numeric or string expression, or an entire program (procedure). A spreadsheet cell could contain the formula format #B3+#C4,

meaning this cell should show the sum of the values in cells #B3 and #C4. However, a cell also can contain the command PERFORM xxx, where *xxx* is the name of a perform file stored on the disk. Then, the cell would take on the returned value of the program.

The value of a spreadsheet cell can be a simple number or string, a numeric or string expression, or an entire program (procedure).

The spreadsheet command line accepts any spreadsheet command or (more important) any valid KMAN command or programming structure. Spreadsheet items, such as auto-recalculation, direction of calculation, and foreground and background colors, are all controlled by environment variables.

The final variable class is field. Field variables can be string (STR) with a maximum length of 65,534 characters, numeric (NUM) with up to 14 digits of accuracy, integer (INT) with up to 14 digits, or logical (LOGIC).

TABLE STRUCTURES

KMAN maintains data in flat tables that are unlimited in number. Each row in a table represents a record; each column a field. In the sample application developed by *PC Tech Journal* to test data managers, information about magazine authors, articles, and issues is kept in separate tables. (For a description of the sample application, see "Sample Application Specifications," August 1985, p. 48. The article also is available for downloading on PCTECHline.) In the KMAN world, each table has a name consisting of up to eight alphanumeric characters followed by the extension .ITB. In the sample case, the tables are named Author.ITB, Article.ITB, and Issue.ITB. Tables automatically reside in a fully encrypted format on disk.

The KMAN command USE opens a given database for use in the KMAN environment—for example, USE Author. Once a table is put into use, the fields of its current record become available as field variables in the programming environment. The table and field names are combined to produce a qualified field variable name. If, for example, the last-name field in the Author table is Lname, the fully qualified field variable name is Author.Lname.

Theoretically, a database comprises the information in one or more tables and the relationships among the information in multiple tables. In KMAN, each table exists as a single DOS file. The relationships between the contents of the files are created by the KMAN user upon requesting data with either the query language at the interactive prompt or commands in a program. A header begins each file, defines the fields in the table, and stores information about the table for the KMAN system. The header information is, in effect, a data dictionary for each table. Unlike other data managers, however, KMAN has no master dictionary containing information about entities, relations, and files (tables) within a database.

A KMAN table can be defined interactively simply by entering the word DEFINE. KMAN prompts for the name of the table and the file in which the table is to be stored, then prompts for entries for up to 255 fields. A field is defined by giving its name, type (STR, INT, NUM, or LOGIC), the size of a STR field, and, optionally, a label and a data picture. Field definition is ended by entering the command ENDDEF.

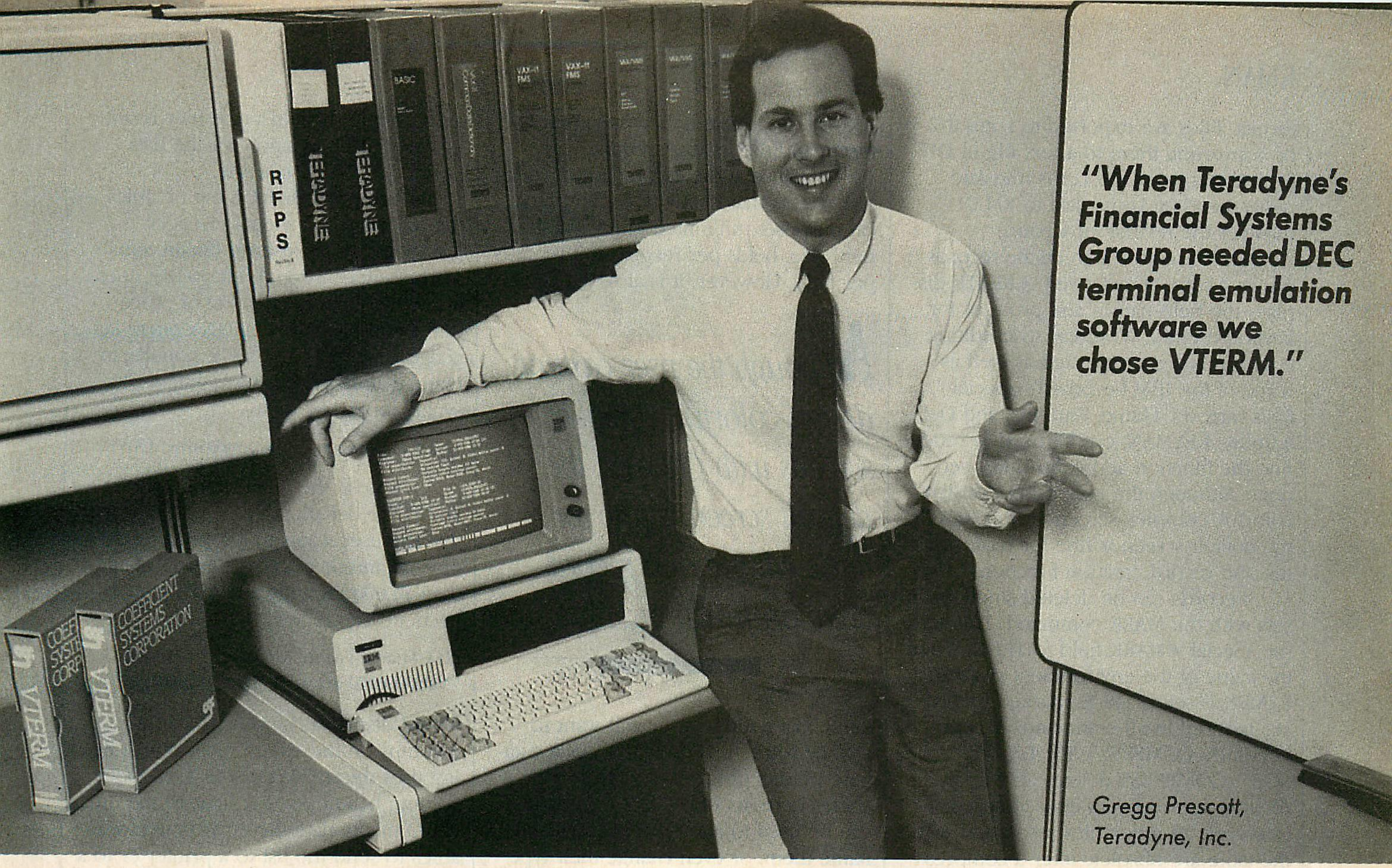
Alternatively, the DEFINE command, the field definitions, and ENDDEF can be put in a procedure file to generate the table in a batch mode. This is especially helpful when dealing with databases that have a large number of fields and complex data pictures.

Besides actual fields, KMAN tables support *virtual* fields. Only the definition of a virtual field is carried in the KMAN table; its value is calculated when needed. Suppose a user wants to see the monthly salary paid to each author on a regular basis even though salary and bonus information are kept only in terms of annual totals. Monthly salary could be defined as a virtual field in the Author table. Its definition might look like the following:

```
Monsal NUM = (Salary + Bonus)/12
  USING "dd,ddd"
```

Author.Monsal is now a legitimate field variable for the Author table. While the Salary and Bonus fields physically exist in the Author table (and take up storage space), the value of Author.Monsal is calculated only when Author.Monsal is used in a KMAN command.

Once a table has been defined, records can be attached from text files in DIF format, BASIC data format, quoted or unquoted ASCII format, or (with a bit of editing) just about any other text



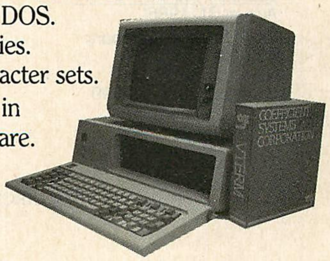
VTERM/220
Quality makes all other DEC terminal emulators obsolete

Over 35,000 VTERM users, like Teradyne's Financial Systems Group, recognize the importance of critically evaluating a DEC terminal emulator. Demanding professionals require high quality, reliable DEC terminal emulation. After painstaking evaluation, Teradyne's Gregg Prescott said, "With VTERM's speed, ease of use, hot key and host control of file transfer, we can build systems around VTERM utilizing distributed PC applications."

Coefficient's VT100 terminal emulator, introduced in 1981, was the first in the industry. Our thorough attention to detail at every stage of the design, development and testing process has won us more satisfied users than all our competition combined. Now, a new more powerful VTERM supports VT220 terminal emulation. Powerful features include:

- Plug compatible VT220 and VT100 video and keyboard emulation with customizable key mappings.
- Optional Tektronix™ 4010/4014 graphics terminal emulation.
- Powerful file transfer including the most thorough implementation of KERMIT available on the PC, plus XMODEM, and our proprietary protocol VTRANS with complete host-side software for VMS,™ RSTS/E,™ RSX11 M/M+™ and UNIX.™
- Host data capture and conversion to Lotus® 1-2-3,® Symphony® and dBase.®
- 132-column display via horizontal scrolling or optional video board.
- Scrollback buffer for redisplay of up to 2,000 lines (80 screens!).
- "Hot Key" toggle between host session and PC DOS.
- Programmable softkeys with script-like capabilities.
- Full support for multinational and national character sets.

Call 212-777-6707 ext. 150, to get the best there is in DEC terminal emulation and communications software.



Coefficient
The Leader in DEC Emulation Software
Coefficient Systems Corporation
611 Broadway, New York, N.Y. 10012

CIRCLE NO. 189 ON READER SERVICE CARD

Trademarks: DEC, VMS, RSTS/E, RSX11 M/M+, Digital Equipment Corp.; Tektronix, Tektronix, Inc.; Lotus, 1-2-3, Symphony, Lotus Development Corp.; dBase, Ashton-Tate, UNIX, AT&T, Bell Laboratories.

format. Files are exported to Lotus 1-2-3 by converting them to a text file in DIF format and using the Lotus translate utility. Sending Lotus data to KMAN involves the reverse procedure.

An alternative is to create records one at a time, interactively. CREATE displays the field names and prompts for keyboard entry. WordStar-style editing commands are recognized as well as some of the IBM keyboard's special keys (arrows, Home, End, Ins, and Del); a utility, KEYMAN, can be used to redefine the function and control key settings. Although the user cannot design colorful, complex screens with CREATE, the default screen is fine for basic editing and record creation functions.

Records can be deleted singly or in sets with the MARK command. This sets the special #MARK field to TRUE. When the value of the environment variable E.IMRK is set to TRUE, marked records are ignored in all commands except UNMARK, COMPRESS, INDEX, and SORT. COMPRESS deletes all marked records and recovers unused space. However, all index files associated with the table are obsolete after a compress and must be rebuilt. If the user wishes, COMPRESS and REINDEX can be added to a shut-down procedure so that whenever KMAN is terminated, all marked records are deleted, their space recaptured, and the indexes rebuilt.

A glance at the accompanying sidebar reveals that KMAN is, for all practical purposes, unlimited in record size, field size, fields per record, and records per field. However, it is almost a certainty that a KMAN user will want to change some field definitions as a database project progresses. KMAN really shines here. Its REDEFINE command can be used to add or delete fields, rename them, or even change their type or size. Obviously, deleting an existing field causes the loss of data in that field, changing the field's size may cause either fill or truncation of existing data, and changing a field's type may result in a new interpretation for the data rather than its destruction.

Although the user cannot design colorful, complex screens with CREATE, the default screen is fine for basic editing and record creation functions.

Getting information out of tables is the business of any data manager. KMAN allows the user to obtain information either through interactive queries or through the creation of procedures and programs. KMAN's query language is similar to SQL/DS, popular in many mainframe database systems. Its syntax looks like this:

VERB expressions FOR conditions SCOPE.

Suppose the Author table is in use and is the default table. Suppose further that the user wants to know the name of each author who both makes more than $4,000 monthly and lives in Connecticut. The following command

```
LIST Lname Fname Monsal FOR Monsal >
  4000 AND State = "CT"
```

would display a list in neat rows and columns on the screen. This default listing uses the utility variable settings for column headings. Alternatively, the user could manipulate the environment variables to send the list to a printer or to a text file. CONVERT could be used instead of LIST to send the information to a disk file in BASIC, ASCII, or DIF format or to another table, or the list could be sent to the interactive spreadsheet for further manipulation.

The LIST command shown above gathers all records that meet the user's conditions. Another KMAN command, OBTAIN, retrieves a single record that meets those conditions. OBTAIN can be used with FIRST, LAST, NEXT, or NEXT(n) to get any particular record in the group that qualifies.

```
OBTAIN FIRST RECORD
  FOR Monsal > 4000 AND State =
  "CT" WITH Fname Lname Monsal
```

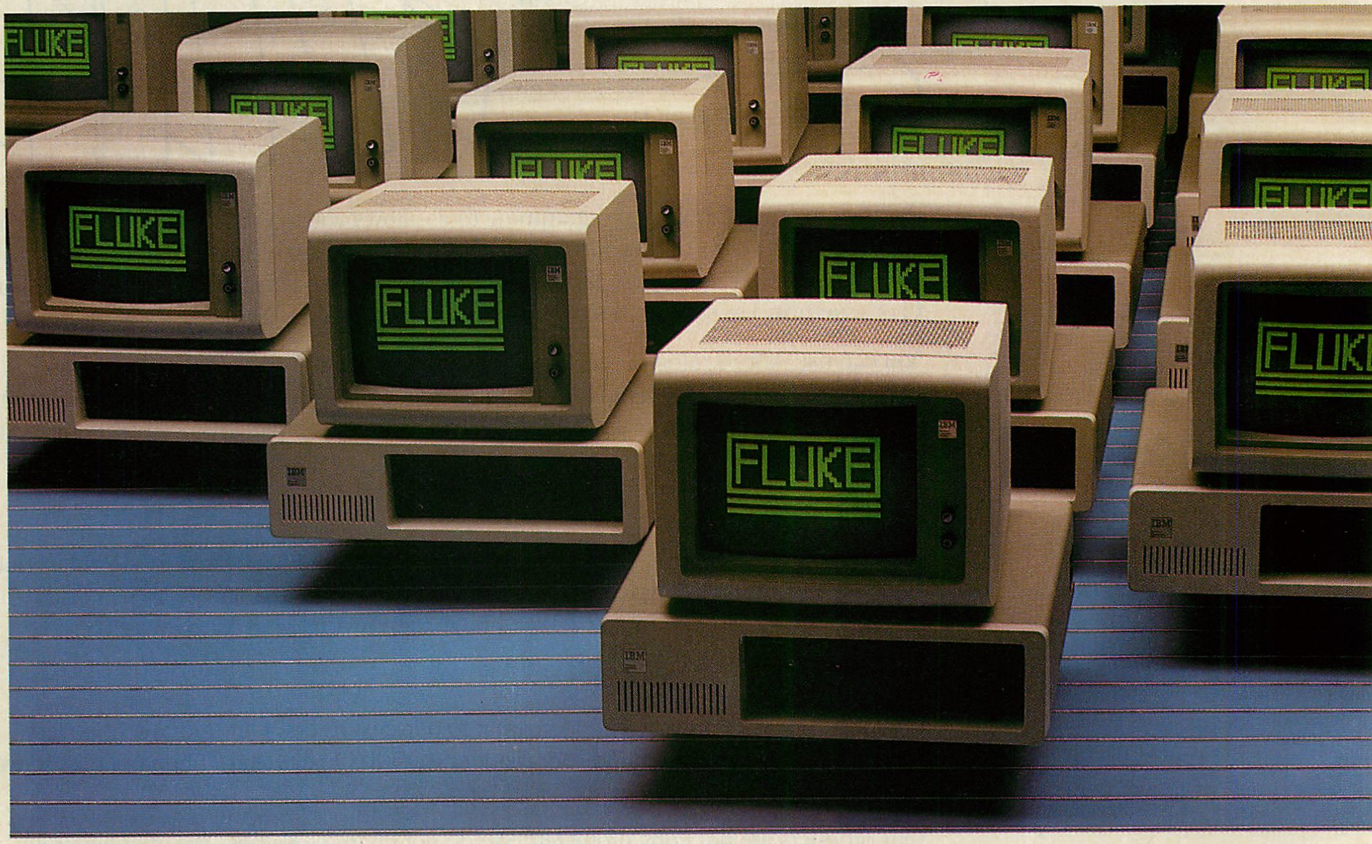
would present the first record in the group in the default edit format for manipulation. If the WITH clause is dropped, all fields are displayed.

If the environment variable E.SUPD (suppress display) were set to TRUE, the selected record would not be displayed, but it would be made the current record, so its field variables would become available for manipulation.

USING MULTIPLE TABLES

If all data could be kept in single, two-dimensional tables, life would be much simpler. But for better or worse, the world is not flat and we must come up with ways of arranging flat data files to model multidimensional situations. Consider the sample application, which

How to keep a whole fleet of IBM PCs up and running.



Introducing Fluke's PC TESTER, the 30-minute turn-key solution for IBM® PC Troubleshooting.

Whether it's a bug in the monochrome board, a sick system board, or a dead disk controller board, Fluke's PC TESTER can reduce the hours you spend troubleshooting an IBM PC to a remarkable 30 minutes or less.

No more messy "rat's nest" of test clips to clutter the bench; just one 40-pin plug inserted into the 8088 socket, and you're up and running a comprehensive set of functional tests on all three boards.

When a bad circuit is detected, PC TESTER's specially designed TECHPRO™ software leads you straight to the bad IC. Concise guided-fault messages tell you which nodes to test with Fluke's signature probe. And PC TESTER can be used by anyone, regardless of experience.

Fluke's PC TESTER consists of a TECHPRO Software package and instruction guide for exclusive use on Fluke's revolutionary 9000 Series Micro-System Troubleshooter along with an 8088 microprocessor interface pod. Prices start as low as $6,490 for the entire system. Software only: $1,500.

So if you're repairing IBM PCs, or thinking about it, take a close look at Fluke's PC TESTER. See how you can troubleshoot a whole fleet of IBM PCs. At a price that makes dollars. And sense.

For more information, call our toll-free hotline, **1-800-426-0361**. Or contact your local Fluke Representative for a free demonstration.

FLUKE®

NOW! Support for the IBM XT and peripherals.

IN THE U.S. AND NON-EUROPEAN COUNTRIES: John Fluke Mfg. Co., Inc., P.O. Box C9090, M/S 250C, Everett, WA 98206, Sales: (206) 356-5400, Other: (206) 347-6100.
EUROPEAN HEADQUARTERS: Fluke (Holland) B.V., P.O. Box 2269, 5600 CG Eindhoven, The Netherlands, (040) 458045, TLX: 51846.

CIRCLE NO. 204 ON READER SERVICE CARD

requires tracking magazine articles and their authors.

A reasonable model of the real publishing world can be made by considering relations among the flat tables. Many articles could have been written by one author. A similar relation exists in that many articles are contained within a single issue.

These many-to-one relationships are handled quite easily by KMAN's multiple table manipulation scheme. However, in the *PC TECH JOURNAL* sample application, more than one author may work on a single article. Therefore, authors and articles can exist in a many-to-many relationship; KMAN cannot handle this situation without some programming help.

In some database systems, links (relations) are defined when tables are created. In KMAN, however, the links are established with the query language or from within a program. For example, the link between the Author and Article tables can be made on the author's first and last name fields. With a many-to-one relation between articles and authors, a list of all articles, their authors, and the author's work phone number would require a query similar to the following:

```
LIST Author.Lname Author.Fname
  Article.Title Author.Wphone
    FROM Article FROM Author
    WHERE Author.Lname = Article.Lname
    AND Author.Fname = Article.Fname.
```

Control-breaks based on the author's name could be added and the environment variables could be set up to route output to various devices.

Working variables and math and string expressions can be included in a multitable list command. As a result, the user is limited only by imagination and time in establishing relations and making queries—as long as either a many-to-one or a one-to-one relation exists between the tables.

Because KMAN allows any number of tables to be in use simultaneously and allows command lines of any length and because relations are established by the query itself, there are virtually no limits to what can be asked or where the question begins. Extraction time, however, places a practical limit of three or four on the number of tables to be queried simultaneously.

Selection time seems to increase geometrically with the number of tables accessed, because the tables are searched sequentially. When index files are used, however, the searches substantially speed up. Indexing can help only when a key value is involved in the search. The good news is that any table can be indexed without practical limits on either the number of indexes per table or the length of the index key. The maximum index key length is 65,535 characters, although MDBS recommends that an index key be no longer than 10 characters for fastest processing. Indexes are maintained in B+trees, and retrievals involving only indexed values are extremely fast.

In addition to its regular working environment, KMAN has its own set of Pascal-like program control structures and commands.

Indexes can be created or destroyed at any time. In order for an index file to be automatically updated along with the table, the file must be mentioned in the USE command when the table is opened, and the environment variable E.INUP, which is used to determine whether or not the index is maintained along with record modification, must be set to TRUE.

PROGRAMMING WITH KMAN

In addition to the working environment already described, KMAN has its own Pascal-like program control structures and commands, including WHILE/DO, TEST/CASE, IF/THEN/ELSE, and PERFORM/RETURN. Working variables can be declared to be local to procedures; otherwise they are considered global variables. Field, cell, and predefined variables are always global. Arguments may be passed to procedures, and procedures may call procedures to an unlimited depth of nesting.

KMAN's powerful macro capability can be addictive. Any eight-character string that begins with an alpha character, except KMAN key words, can be used to name a macro. The contents of the macro can be of any length—an entire program if the user wishes; furthermore, macros may call other macros to an unlimited depth of nesting. This facility can be used to modify the command language, to create user-defined functions, to take care of repetitive verbiage during a retrieval session, to execute a series of commands used in creating a report, or even to make the

use of environment variables somewhat more understandable.

Consider these macro definitions:

MACRO printon E.OPRN = TRUE
MACRO printoff E.OPRN = FALSE

These two macros create two new KMAN commands. The first, **printon**, starts output to the printer, and the second, **printoff**, stops it. This use of macros can make KMAN programs much more readable.

As mentioned earlier, procedures and groups of statements and commands are kept in .IPF files until called by the PERFORM or INCLUDE commands. Normally, once a procedure finishes executing, its memory is freed; however, procedures invoked by the LOAD PERFORM command remain in memory for reuse until freed by the RELEASE command. Procedures can be combined into a library through the Library Building utility (KLIB). This approach can save procedure access time and reduce the number of open files. Once the library is loaded, it stays memory-resident until released.

KMAN neither handles transaction processing nor creates audit trails automatically. These functions are left up to the programmer to resolve.

Because of its modular structure and interpreter environment, KMAN is fairly simple to debug. Usually a procedure is developed and debugged interpretively, then stored as an .IPF file to be called later. Environment variables, including E.STEP (display each statement before executing), E.PAUS (pause when the screen is full), and E.SERR (display or suppress error messages), along with the I/O control E-variables, can be used to diagnose problems. With almost 150 error messages that can be trapped, displayed at runtime, or recorded to disk or printer, the error handling routines can be as complex as the designer wishes.

Programming in the KMAN operating environment is comfortable. The RUN command loads the DOS command processor to allow the user to enter DOS commands or run other programs. Typing EXIT at the DOS prompt returns the user to KMAN. Alternatively, RUN can be followed by a command line that places the user directly into an external program. In this case, the user is returned to KMAN upon termination of the external program. This feature allows the programmer to use a favorite text editor for program development or perform other work without disturbing the KMAN set-up.

KMAN also features its own directory list, file rename, and file destroy routines that can be used in a procedure or program without ever leaving the KMAN environment.

MDBS offers a runtime package for commercial developers. The package includes an encryption utility, called SCRAM, which can be used to keep source code secure. All data tables are automatically encrypted.

dBASE-LIKE FORMS

Screen I/O can be as simple or sophisticated as the user wishes. The commands BROWSE and CREATE (record) use a default screen that is a vertical column of field names followed by field values, if any. These screens enable full screen editing, so they are appropriate for many ad hoc applications.

For commercial applications, the base program includes the FORM command, which allows complex forms to be created interactively or in a procedure file. KMAN's form system is similar to Ashton-Tate's dBASE series in that a form is composed of a number of statements that provide the screen coordi-

FIGURE 1: *Sample Forms*

```
/* TPFORMS.IPF   */
FORM AUTMEN
  AT 2, 25 PUT "PC TECH JOURNAL DATABASE PROJECT"
  AT 3, 28 PUT "Sample Magazine Inventory"
  AT  8,16 PUT  "A> Add an author" WITH "L"
  AT 10, 16 PUT "B> Edit author tables" WITH "L"
  AT 12, 16 PUT "C> Return to Main Menu" WITH "L"
  AT 14, 16 PUT "D> Return to DOS"  WITH "L"
  AT 20, 16 PUT "Please select a LETTER ==>"
  AT 20, 59 GET CHOICE STR  USING "u" WITH "R"
ENDFORM

FORM AUTED
  AT 6, 1 PUT "Social Security Number:"  WITH "L"
  AT 6, 25 GET Ssn NUM USING "nnn-nn-nnnn"
  AT 6, 25 PUT Ssn USING "nnn-nn-nnnn"
  AT 9, 6 PUT "First Name:" WITH "L"
  AT 9, 18 GET Author.Fname STR USING "%12a"
  AT 9, 18 PUT Author.Fname USING "%12a"
  AT 9, 41 PUT "Last Name:" WITH "L"
  AT 9, 52 GET Author.Lname STR USING "%18a"
  AT 9, 52 PUT Author.Lname USING "%18a"
  AT 11, 6 PUT "Street address:" WITH "L"
  AT 11, 36 GET Author.Address STR USING "%20r"
  AT 11, 36 PUT Author.Address USING "%20r"
  AT 13, 6 PUT "City:" WITH "L"
  AT 13, 12 GET Author.City STR USING "%16r"
  AT 13, 12 PUT Author.City USING "%16r"
  AT 13, 31 PUT "State:" WITH "L"
  AT 13, 38 GET State STR USING "uu"
  AT 13, 38 PUT State USING "uu"
  AT 13, 43 PUT "Zip:" WITH "L"
  AT 13, 48 GET Author.Zip NUM USING "nnnnn"
  AT 13, 48 PUT Author.Zip USING "nnnnn"
  AT 15, 6 PUT "Work Phone:" WITH "L"
  AT 15, 23 GET Author.Wphone NUM \
                USING "(nnn) nnn-nnnn"
  AT 15, 23 PUT Author.Wphone \
                USING "(nnn) nnn-nnnn"
  AT 15, 41 PUT "Home Phone:" WITH "L"
  AT 15, 53 GET Author.Hphone NUM \
                USING "(nnn) nnn-nnnn"
  AT 15, 53 PUT Author.Hphone \
                USING "(nnn) nnn-nnnn"
  AT 17, 6 PUT "Biography:" WITH "L"
  AT 18, 6 GET Author.Bio STR
  AT 18, 6 PUT Author.Bio
ENDFORM
```

A form is composed of a number of statements that provide the screen coordinates for text, prompts, and data. KMAN's form system may be used to build menus, data entry screens, and some reports.

nates for text, prompts, and data. Two typical forms are shown in figure 1.

Note the similarity to a dBASE form (the AT key word can be replaced by an @ character). The PUT command writes to the screen. Control codes can be sent to the printer with the PUT command using the CHR function. The GET command, together with GETFORM, reads input from the screen. Data can be displayed according to picture clauses and with what MDBS calls special effects: low intensity, blinking, reverse video, or sounding a bell when the cursor enters the field. In figure 1, the phrase WITH "L" causes the value to be displayed in low intensity. Additionally, rectangular blocks of color can be defined by giving the northeast and southwest coordinates.

After the constant fields of a form are displayed with PUTFORM, the TALLY command can be used to fill in the values of data items. After input is entered, TALLY recalculates values that are dependent on screen input. If the environment variable E.ICOM is set to TRUE, values are recalculated automatically as input is entered.

Several forms can be on the screen simultaneously, and multiple tables can be accessed within a form, but only one record from each table can be displayed at a time. This simplifies the challenge of updating multiple tables with what appears to be a single screen.

PICTURE EDITING

Most of KMAN's automatic editing is derived from the definition of the picture for a data item. A *picture* is a sequence of characters that provides a mask for data I/O. Typically, pictures are defined in USING clauses. A field variable can be given a default picture when the data table is defined. Any later picture that is given in the GET and PUT commands for a form temporarily overrides the default value.

With pictures, data can be shifted automatically, upon input, to upper- or lowercase or, perhaps, to leading caps only. Data elements that usually have special formatting characters can be stored without those characters to conserve space; the special characters are added only when the data are displayed. For example, a phone number displayed as (404) 555-1212 is actually stored as 4045551212. The phrase to create the appropriate output formatting or input mask on a screen is:

... USING "(nnn) nnn-nnnn"

The KMAN placeholders used in defining a picture follow:

a—any alphabetic character
c—any alphanumeric character
l—any ASCII character; uppercase alphabetic characters are converted to lowercase
u—any ASCII character; lowercase alphabetic characters are converted to uppercase
r—any ASCII character; no case conversion
n—any digit
d—a digit, sign (+ or −), or decimal point
f—behaves like **d**, except that literal characters to the left of the first f float to the right until a non-f placeholder is encountered; this

placeholder is used to position a dollar sign in front of the first significant digit of a number

s—behaves like **d**, except literal characters (other than decimal point) that appear between the first and last **s** of the picture are replaced with blanks if a digit does not appear on both sides

Placeholders must be in lowercase. Any other characters in a picture are assumed to be constants. The placeholder characters can be used as constants by prefixing them with two backslashes. The % symbol followed by a count may be substituted for a sequence of identical placeholders. For example, %**5u** and **uuuuu** would create the same data picture.

Notice that a limited amount of editing is done automatically, such as restricting input to digits or alphabetic characters. The **u** and **l** features make testing for lower- or uppercase answers to a prompt unnecessary. Range checking, verification of the input against lists of valid values, and the use of look-up tables must be accomplished with programming. For example, the clause USING "nn" restricts input to two digits, but it does not control which two digits.

Additionally, with programming and special effects, the display attributes can be varied depending on the value of the data. For example, code can be written to determine if a variable has a positive or negative value; then the variable might be displayed in green if positive or red if negative.

Range checking, verification of the input against lists of valid values, and the use of look-up tables must be done with programming.

K-Paint is a utility that, as its name implies, allows the user to paint a screen interactively (with or without a mouse). This is convenient for large, complicated jobs, especially when the general screen/menu style of a given application is repetitive. K-Paint generates an .IPF file that includes the FORM, ENDFORM, AT *x,y*, GET, and PUT clauses. It does no automatic programming. Range checking, linking forms together, and manipulating data within the form are up to the programmer.

When PAINT (form name) is invoked from KMAN's command line prompt, the screen is cleared except for a two-line menu/status area at the bottom. The menu choices in themselves create a logical approach to screen design. The first choice is whether to work on a color block or an element. If color block is selected, the user is instructed to put the cursor in the northwest corner of the block (screen area), press Enter, move the cursor to the southeast corner, press Enter again, and then select foreground and background colors from a list on the command line. If element is selected, the user is instructed to move the cursor to the beginning of the element (on the screen) and to choose literal or nonliteral. For literal, the program prompts the user to type the literal—perhaps a prompt such as "Enter your first name:" The command line then asks for attributes for the literal, such as blinking, low intensity, or a bell that is to be sounded when the literal is displayed.

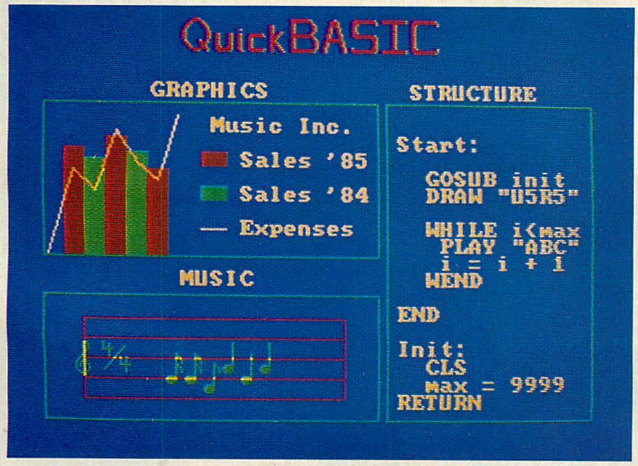
A user who elects to work on a nonliteral is requested to identify a variable or an expression and to assign it a location and attributes. The user continues in this manner, identifying literals, nonliteral expressions, and their locations and attributes until the form is

Get up to speed for under a hundred bucks.

If you're writing in the BASIC that came with your IBM® PC, now you can make your programs scream. Microsoft's new QuickBASIC Compiler will punch you right into warp speed, accelerating your programs by as much as ten times. And all with hardly any code changes.

You get all this at a price that won't slow you down, either. Just $99 and you're in.

The new QuickBASIC Compiler was designed to work with the



programs you've already written. It supports all BASIC sound and graphic statements including PLAY, SOUND, LOCATE, DRAW, GET, PUT, LINE, CIRCLE, PSET, and COLOR.

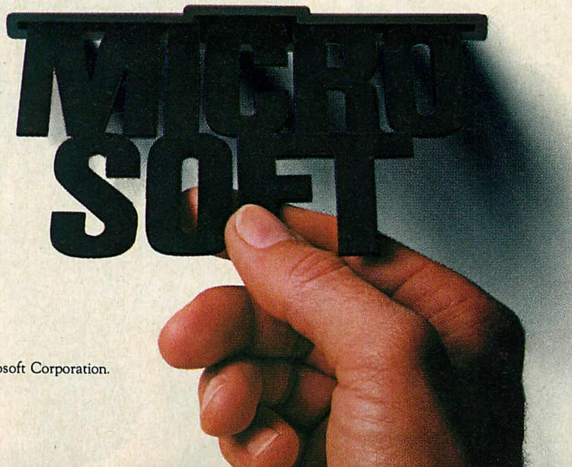
So your graphics and music routines will look and sound as they should.

And when you're ready to move into structured programming, QuickBASIC also offers language extensions for a speedy transition. With optional alphanumeric labels you can make your programs more readable. And compiled subprograms will save you time.

MICROSOFT®
The High Performance Software™

So get moving. Call us now for more information on the QuickBASIC Compiler and the name of your nearest Microsoft dealer. Just ring (800) 426-9400. In Washington State and Alaska, (206) 882-8088. In Canada, call (416) 673-7638.

Then go give your programs a swift kick.



ZIM 2.5 A DBMS REVOLUTION

Have you been looking for perfect data management that you can enjoy on your own terms? Then you've probably already heard of ZIM 2.4 — the most powerful database system available. Until now. Because ZIM 2.5 is here.

ZIM 2.5 is a fourth generation application development tool which makes it possible to expand the capabilities of your micro beyond what you've ever imagined. ZIM mirrors the complexities of the real world by letting you develop as many and as varied applications as you could possibly need.

"ZIM is...a successful migration of mainframe ideas and needs to a micro. (ZIM) proves not only that the job can be done but also that it can be done well. ZIM provides a reference against which current and future data bases can be judged." James Creane, Data Based Advisor/July 1985.

Speed
ZIM breaks the speed limit — between 3 and 50 times faster than industry leaders in sorting and joining files within the data-base. ZIM's internal architecture, and the implementation of its strategy analyzer and priority-driven buffering ability, ensure that data is processed in the most efficient manner possible.

Portability
ZIM is the only database management system with 100% application portability for single-user and multi-user configurations. ZIM is available under PC-DOS, Concurrent PC-DOS, UNIX, XENIX, and QNX. Never again will you be required to re-write your applications for different operating systems environments.

Power
ZIM's high-level language lets you build user commands which implement applications without the necessity and cost of additional programming tools. ZIM's forms facility and extensive report generator permit completely menu-driven applications. Completed compiled, applications use the Runtime System, leading to fast execution, preventing unauthorized access or modifications, and decreasing cost and memory requirements.

Flexibility
ZIM gives you unprecedented simplicity and flexibility. ZIM commands parallel simple English sentences, making it easy to learn and use. Other features include automatic updates of all indexes, multi-user support, and an extensive range of validation, editing and masking facilities. ZIM's limits are only those of your hardware, operating system and imagination. And with ZIM 2.5, your database is no longer limited to a single hard disk.

"ZIM is (a) well-conceived, soundly-implemented, thoroughly professional system. Its design evidences a strong commitment to consistency and to the goal of natural nonprocedural user interaction."

Richard M. Foard, PC Tech Journal, October 1985.

ZIM 2.5 — DATA MANAGEMENT AT ITS BEST

38 Antares Dr., Suite 1200
Nepean, Ontario, Canada
K2E 7V2
(613) 727-1397
ZANTHE
INFORMATION INC

CIRCLE NO. 219 ON READER SERVICE CARD

completed and ready for use. Color blocks and elements can be moved around the form easily.

Forms can be as large as 255 lines by 255 columns. When the K-Paint user moves the cursor outside the screen's viewing window, the window is shifted automatically to the previously invisible part of the form.

VARIETY OF REPORTS

One way to generate a report is to create a form (with or without the K-Paint module). The simplest type of report is generated with the LIST, SELECT, OBTAIN, or CONVERT commands. Any one of these can direct the output to the printer, screen, or disk as desired by the user. However, KMAN provides several other ways of generating reports of varying complexity.

The LIST and SELECT commands provide a columnar listing of field and expression values for each record in one or more data tables that meet the selection criteria. Utility variables can be used to create a header for the report and to replace field names with more descriptive heads above each column in the listing. Environment variables control page ejects, pagination, margins, and printer start/stop control sequences. LIST and SELECT cause automatic wrapping if columnar output exceeds the margin values that are set in the environment variables.

If the environment variable E.STAT is set to TRUE, statistics on the report's columns are generated automatically. The statistics listing for string expressions shows the count and minimum and maximum values. Statistics for each numeric expression include count, sum, average, minimum, maximum, variance, and standard deviation. The only statistic given for logical expressions is the count. Environment variables can be used to turn off all statistics or individual items in the statistics list. (The STAT command is discussed later.)

OBTAIN provides a simple listing of selected field variables belonging to a single record. This command or the KMAN form with some programming can be used to produce the one-page-per-record type of report.

CONVERT can be used for report generation in programmed applications to produce one or more lines of data in ASCII format from records that meet certain criteria. One of the main applications of this command is to create mailing list output to the printer.

Yet another way of creating a report with KMAN is to use the LIST, SELECT, OBTAIN, or CONVERT statements together with programming language elements. This produces a report-making program that typically selects a record meeting certain criteria, outputs data to be printed, and keeps track of subtotals and totals for later printing. Such a program usually includes a page numbering routine and generates its own headers and footers. This type of report generation must be used if more than one table is queried and a many-to-many relation exists between the table elements.

KMAN provides an optional utility called K-Report that saves time in designing complex reports, although the reports have to be quite complicated before much time saving is realized. This is the case primarily because K-Report (as of this writing) suffers from several bugs. KMAN version 2.0 uses an entirely different screen-handling routine than did earlier versions, and K-Report has not yet caught up with the changes. Its column counter is not always correctly incremented. Like K-Paint, K-Report handles reports larger than can be displayed at one time. However, as the user scrolls to another viewing area, the screen image often is corrupted. Positioning to the northwest corner and repainting the screen is necessary to restore the correct image.

MDBS is working on fixes for K-Report and is considering a report generation language because it has been requested by many users.

Another problem with K-Report is that it cannot handle many-to-many relations among tables. Output in a situation involving many-to-many relations can be programmed so it is directed into a temporary KMAN data table. Once this is finished the user can forget about relations and use K-Report in order to generate a report from the temporary table.

K-Report looks like and is used in the same manner as K-Paint. K-Report is invoked by the command DESIGN <report name>. The user begins with a blank screen that has only a menu/status area at the bottom. The choices are to work on a detail pattern or headers and footers for a group, page, or report. By following the prompts, these patterns are developed into an on-screen template that is used by the system to generate reports.

After the template is designed, template data are kept in a file called ReportName.TPL. Reports are generated with the command:

```
REPORT "template filename" selection
  criteria
```

The selection criteria phrase is similar in syntax to LIST or SELECT.

Eleven levels of grouping are permitted in templates. Pagination is automatic with page numbers appearing in page headers or footers as desired. Therefore, K-Report is particularly good at a report that has, for example, company, division, and office headers followed by employee detail records. Subtotals can be calculated by office and division, and company totals given. Here, the level depth is only four.

DATA SECURITY

KMAN's security is among the best in its class. Data can be secured from read or write access by table or field variable. When a table or field is defined, it may be given any combination of 16 read and 16 write access codes. Users also may be assigned any combination of 16 read and 16 write access codes by the USRMAN utility. A table can be accessed only if a user's access code matches one of the table's. A user may access a field only when one of the field's access codes matches one of the user's.

A user who is denied access to a field that is later used in an expression also is denied access to the results of the expression. For example, a user who is denied access to the Salary field in the Author table is automatically denied access to the virtual field Monsal as well, because it is calculated using the Salary field. Spreadsheet cells can be protected against modification. This is a blanket restriction and not based on user access code.

When KMAN first appeared on the market in May 1983, it was criticized for being "user hostile." The initial documentation, while more or less complete, was absolutely sterile and assumed that the user understood (1) the theory of relational databases; (2) the theory of structured programming; and (3) the design intentions of MDBS.

KMAN revisions 1.04 through 1.07 were largely concentrated on getting all of KMAN's features working in concert. Version 2.0 (KMAN/2) is a plea for understanding and acceptance from the common man. Source code in perform files created on earlier versions of the product is generally upwardly compatible, although some programmers have elected to rewrite some stretches of code, thus taking advantage of certain new functions and constructs.

Powerful MS™-DOS Software.

For the IBM® - PC, XT, AT & others with generic MS-DOS/PC-DOS 2.0 or higher.

Not copy protected!

Only **$39.95** Each!

UTAH COBOL™

Whether student, teacher or professional programmer, this is the one you've heard so much about.

- ☐ It's easy to use. Compiles 5000 statements on a 128K machine.
- ☐ 170 clear error messages, i.e. DATA-NAME IS MISSING OR MISSPELLED.
- ☐ Distribute your object code programs royalty free.
- ☐ Small object code programs conserve disk space.
- ☐ Fast compile times to increase programmer productivity. Over 25 times faster than one compiler costing $995!
- ☐ You get a diskette and 213-page manual with lots of examples and 16 complete COBOL source code programs. **$39.95.**

Also available: COBOL Application Packages, Book 1 **$9.95.**

UTAH PASCAL™

- ☐ 14-digit precision, BCD math, no round-off errors with decimal arithmetic for business and floating point +63 −64 for scientific.
- ☐ A very nice TRACE style debugging.
- ☐ Arrays up to 8 dimensions and 64K strings.
- ☐ External procedures and functions with dynamic auto-loading.
- ☐ One-step compile, no assembly or link required.
- ☐ You get a 132-page manual and diskette. **$39.95**

UTAH PILOT™

- ☐ Perfect for industrial training, office training, drill and testing, virtually all programmed instruction, word puzzle games, and data entry facilitated by prompts.
- ☐ John Starkweather, Ph.D., the inventor of the PILOT language, has added a built-in full-screen text editor, and much more.
- ☐ Meets all PILOT-73 standards for full compatibility with older versions.
- ☐ You get a diskette, 125-page manual and ten useful sample programs. **$39.95.**

Also still available for 8-bit machines with CP/M® is our world famous Nevada Software Series used by 50,000 customers in 40 countries. These include Nevada COBOL, Nevada FORTRAN, Nevada PASCAL, Nevada PILOT, Nevada BASIC and Nevada EDIT. **$39.95** each.

Satisfaction guaranteed. If for any reason you're not completely satisfied, just return the package within 15 days in good condition, and we'll refund your money.

UTAH FORTRAN™

- ☐ FORTRAN IV based upon ANSI-66 standards.
- ☐ Very fast compile times and easy to use.
- ☐ IF . . THEN . . ELSE constructs.
- ☐ Chaining with blank and named common.
- ☐ Copy statement.
- ☐ ENCODE and DECODE.
- ☐ Free-format input and output.
- ☐ A very nice TRACE style debugging.
- ☐ 150 English language error messages.
- ☐ You get a diskette, and 223-page manual. **$39.95**

UTAH EDIT™

- ☐ A character-oriented full-screen video display text editor designed specifically to create COBOL, FORTRAN and PASCAL programs.
- ☐ Only requires 15K disk space so it can fit on the same disk as your compilers.
- ☐ Completely customizable tab stops, default file type, keyboard control key layout and CRT by menu selection.
- ☐ Diskette comes with easy to read 58-page manual. **$39.95.**

UTAH BASIC™

- ☐ This interpreter has a built-in full-screen editor.
- ☐ Single- and Multi-line user definable functions.
- ☐ BCD Math- no round-off errors.
- ☐ Full Matrix operations.
- ☐ You get 220-page manual and diskette. **$39.95.**

Handling/Shipping: No shipping charge within US. Overseas add $10 for first package, $5 each additional. Checks must be in US Dollars, drawn on a US bank.

Utah Software requires 128K RAM and PC-DOS or MS-DOS 2.0 or higher.

HOW TO ORDER. Send check or money order to Ellis Computing, Inc. with VISA or MASTERCARD order by phone. Sorry no COD's.

Ellis Computing, Inc.
5655 Riggins Court, Suite 10
Reno, Nevada 89502
Phone (702) 827-3030

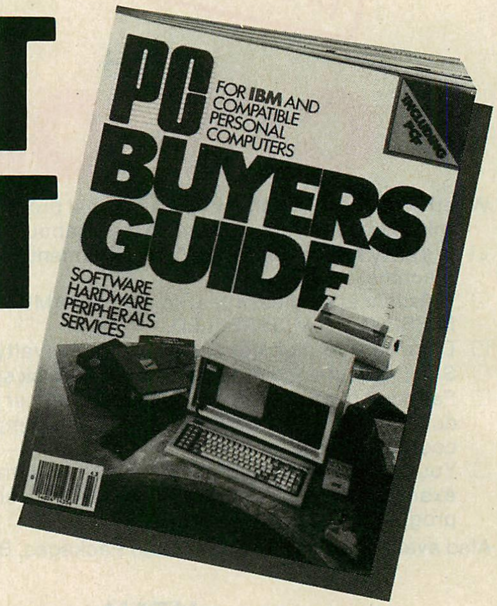
SINCE 1977

ELLIS COMPUTING™

CIRCLE NO. 170 ON READER SERVICE CARD

BEFORE YOU BUY ANYTHING FOR YOUR IBM PC,

CONSULT THE EXPERT



Software, hardware and accessories are not purchased on impulse. With so many products introduced on the market, it's necessary to have the most current reference source available to assist you in making smart buying decisions—

That's why you should comparison shop with PC BUYERS GUIDE!

PC BUYERS GUIDE is the master directory of products for IBM and compatible personal computers. It's more than 500 pages include listings for over 10,000 products, organized in one easy-to-use, comprehensive guide.

Discover what's available, suitable and affordable in software, hardware and peripherals for your IBM PC, PCXT, PC AT, PCjr and compatible personal computer.

Updated and published quarterly, *PC BUYERS GUIDE* is a veritable encyclopedia of data incorporating the newest releases and providing accurate product descriptions and costs—PLUS *PC BUYERS GUIDE* is cross-indexed and categorized so you can easily locate any listing!

Compiled by Data Sources, the most extensive database maintained in the entire marketplace, *PC BUYERS GUIDE* guarantees you the most current reference sourcebook of its kind!

Order your copy today!

To order your copy of the most recent edition complete the coupon below, enclose payment and mail!

Mail to: **PC BUYERS GUIDE** Ziff-Davis, One Park Avenue, 4th Floor, New York, NY 10016, K. Armstrong

Enclosed is my check or money order for $9.45 ($7.95* plus $1.50 for postage and handling) for each copy. (Price outside U.S. $11.00) Enclosed is $__________. Please send ________ copies of the latest edition of PC BUYERS GUIDE.

Mr./Mrs./Ms.____________________
(please print)

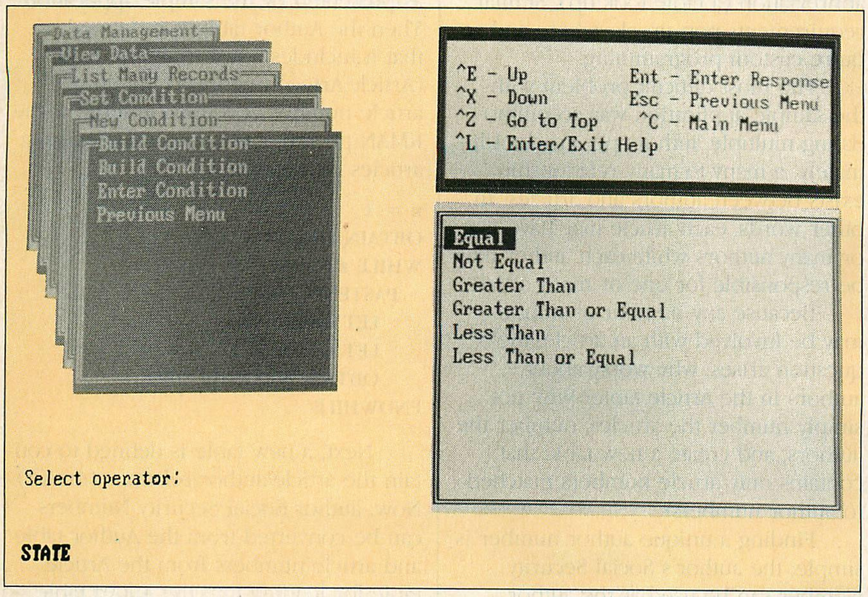
Address____________________

City/State/Zip____________________

AN44

*Residents of CA, CO, CT, DC, FL, IL, MA, MI, MD, NJ, NY State, and VT add applicable sales tax.

PHOTO 1: *KMAN Menu Interface*



KMAN/2 introduced a menu interface for the novice user. The series of menus and prompts takes a user step by step through building a command or query.

Documentation used to be one of KMAN's weakest points. With version 2.0, however, documentation—both printed and on-line—is one of its real strengths. The original documentation, consisting of a single, four-inch thick, 8½-by-11-inch, three-ring binder, has been replaced with two volumes (of the standard IBM manual size and type) of typeset material. The *Reference Manual* is simply a reprint of the old documentation, meant for users of the earlier versions. The additional *User's Guide* is tutorial and presents the capabilities of KMAN in an understandable way.

More important, KMAN now has extensive on-screen help available both in context-sensitive text and in a new menu interface. The applications designer can elect to enable or disable the help and pull-down menus; a newcomer to KMAN, however, will want to start with the menu interface in place.

The menus take the user sequentially through the decision process to get a task done. The menu interface divides the screen into five areas as shown in photo 1. The area in the upper left corner tracks the path the user takes to get to a particular stage. As a selection is made from each menu, it is printed on the top line, and a new menu is presented in front of the old. The utility area on the right is used to provide a guide to function keys and to display available options. The message area in the lower left is used to display prompts to the user; below that are the user entry area for text (not shown in the photo) and the command area, which holds the KMAN command built through the selection of menu choices.

Photo 1 captures a step in building the query of the Author table for all authors who live in California. The menu selections made were Data Management, View Data, List Many Records, Set Condition, New Condition, and Build Condition. Then a menu of field names was presented in the utility area from which the field STATE was chosen. The next step would be to choose the condition the State field must meet. In this way, a query is built step by step until it is ready to be executed.

REAL WORLD KMAN

The editors at *PC Tech Journal* specified an application for this series of data manager reviews, which includes some ad hoc queries, reports, and performance benchmarks. The application is a magazine inventory management system—a set of routines that tracks articles, authors, and magazine issues.

For the KMAN programmer, the sample application presents an interesting challenge—not so much how to do it, but where to stop. Nearly all of the specifications can be handled by KMAN with commands entered from the prompt in the KMAN environment.

KMAN has no limits on the number of files that can be open simultaneously. Additionally it has no limit to the length of a command line, and all KMAN commands, variables, and control structures can be used at anytime, anywhere. The only possible impediment is KMAN's inability to deal automatically with many-to-many relationships.

What about input editing? A large part of that can be handled at the command level as well. For example, many of the editing specifications require that a field contain only alpha or only numeric characters or be a certain length. All of these requirements can be handled when the data table is defined by using picture clauses. In this case, data going into the table from any source (such as imported ASCII file or BROWSE or CREATE from the prompt) are edited automatically.

While this capability is a big help, it does not provide for range checks, checking for duplicate values, verification against lists of valid values, and required fields. In the sample application, entries in the first and last name fields are required for each author record. In the Article file the Category field must contain one of four possible entries: Product Review, Technical Article, Department, or Tech Notebook. Another requirement is that the State field in the Author table be a valid two-letter abbreviation (a table look-up). Similar requirements pop up elsewhere and require custom programming.

The most difficult problem with the sample application was accommodating multiple authors per article. Ultimately, a many-to-many relationship exists between authors and articles. In other words, each article may have one or many authors while each author may be responsible for one or many articles.

Because any number of authors may be involved with an article, the question arises: why worry about authors in the Article table? Why not simply number the articles, number the authors, and create a new table that contains only article numbers matched to author numbers?

Finding a unique author number is simple; the author's Social Security Number can be used as the author number (Author.Ssn). Articles do not have Social Security Numbers, so a unique article number (Article.Artno) has to be assigned to each existing article, and new entries must automatically be assigned numbers.

KMAN's table manipulation facilities allow it to be used for the redesign of the data structures and the data elements in the following way. First, the Author table is defined and loaded just as presented by the sample application. Then the Author table is redefined so that it includes a new numeric field (Article.Artno) to contain the unique article number. A routine entered at the KMAN prompt can number the existing articles (a one-time job):

```
x = 1
OBTAIN FIRST RECORD
WHILE #FOUND AND NOT
   PASTEND(Article)
      LET Artno = x
      LET x = x + 1
      OBTAIN NEXT RECORD
ENDWHILE
```

Next, a new table is defined to contain the article/author relations (AutArt). Now, author Social Security Numbers can be converted from the Author table and article numbers from the Article table for loading into the AutArt table.

Finally, four fields are removed from the Article file: author last name, author first name, coauthor last name, and coauthor first name. This information is no longer needed in Article because AutArt now provides a path for programming a link between articles and authors (and in the other direction if the user desires).

KNOWLEDGEMAN/2 OVERVIEW

KNOWLEDGEMAN/2
Micro Database Systems, Inc., P.O. Box 248, Lafayette, IN 47902; 317/463-2581

Product description. KnowledgeMan/2 is an applications development system with a relational database system, a structured programming language, an SQL-like query language, and an integrated spreadsheet. The product can be expanded modularly to include business graphics (K-Graph), text processing (K-Text), report generation (K-Report), interactive forms design (K-Paint), mouse support (K-Mouse), C language interface (K-C), and communications (K-Comm).

IBM PC environment. KnowledgeMan/2 runs on any IBM PC family computer running under DOS with 320KB memory. A hard disk is recommended but not required. The availability of 1MB of disk storage also is recommended. The product runs with Windows, DesQ, GEM, or TopView.

Other environments. The multiuser version runs under UNIX on VAX-11 series computers, as well as under the CP/M-86 operating system.

Network support. Networks supported include Novell's NetWare, 3-Com's EtherShare, and the IBM PC Network.

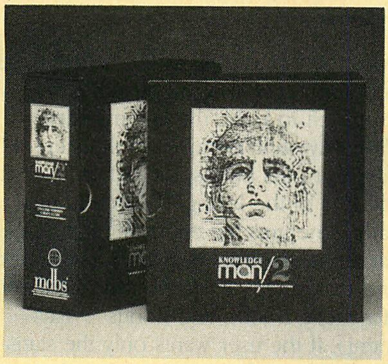
Copy protection. KnowledgeMan/2 is not copy protected.

Documentation. Extensive documentation is provided for the novice in the tutorial *User's Guide* and for the advanced user in the *Reference Guide*. Each optional module comes with its own documentation and section tab designed to be inserted into the existing IBM-size volumes.

User interface. KnowledgeMan/2 introduces a new menu interface with prompts that guides the novice through the process of building commands or queries. This interface may be disabled, allowing the advanced user access to the interactive command mode. Macros can be defined to facilitate repetitive execution of long commands or to make KnowledgeMan/2 programs more readable.

Help facilities. Comprehensive general and context-sensitive help is available on-line. Optionally, upon detection of a syntax error, KnowledgeMan/2 displays the error message and offers the user further help.

File capacities. File capacities are virtually unlimited. The maximum



number of files in a database and the maximum number of tables open simultaneously is limited only by the operating system. An unlimited number of index keys per table are allowed with a maximum size of 65,535 characters per key and 65,535 fields per index. The maximum number of records per file is 2,147,483,647, with 255 fields per record up to 65,535 characters combined.

Field types and capacities. Fields can be defined to be one of four types: string (STR) with up to 65,534 characters, numeric (NUM) with 14-digit accuracy, integer (INT) with up to 14 digits, or logical (LOGIC). The use of arrays is not supported for fields.

Data entry. Data pictures can be specified for fields when the table is defined and overridden by another picture in a data entry form. These pictures are effective for limited editing, such as restricting input to numeric or alphabetic characters only. More extensive editing for default values, table look-ups, range of values, and required fields must be programmed.

Application development facilities. The programming language features elements such as looping blocks, nested procedures to an unlimited depth with parameter passing, and global and local variables. An unlimited number of working variables is supported, and working variables can be arrays. Command lines and procedures can be any length. Functions are available for numeric calculations and string processing. Environment and utility variables can be set to control formats and configure output. Some utility variables hold status information reflecting current database activity. Data can be moved easily between tables and spreadsheets.

Security. Access to tables and to fields is handled through read and write access codes matched to those assigned to users. Spreadsheet cells can be write-protected. Tables are automatically encrypted, and a system utility, provided with the runtime module, is used to encrypt procedure files.

Access to system facilities. The DOS command processor can be invoked either for one command or for execution of an external program. KnowledgeMan commands are available for directory listings as well as for file renaming and deleting.

Query and sorting. The product's query language is based on SQL (Structured Query Language). Multiple tables can be queried with output directed to the screen, a disk file, or the printer. Statistics are computed automatically, including count, minimum, maximum; and for numeric fields, count, average, variance, and standard deviation.

Reporting. Without the optional K-Report package, most reports have to be programmed.

Utilities. Utilities are available for selecting a driver for different terminal types and for customizing the keyboard. With the USRMAN utility, a master user assigns access codes to other users. Additional utilities are provided to consolidate overlays into the main KnowledgeMan/2 program and to combine multiple procedure files into a library.

Data compatibility. Data can be input from and output to files in ASCII delimited, BASIC, and DIF formats.

Distribution. Began in May 1983 through distributors and dealers.

Price. $595; upgrade for existing users, $295; LAN version, $1,795 (10-user maximum), $3,325 (32-user maximum); K-C, $1,175; K-Comm, $225; K-Graph, $225; K-Mouse, $100; K-Paint, $100; K-Report, $225; K-Text, $175. Runtime versions are available: K-Run for KnowledgeMan/2 and K-RunX for the optional modules are sold for $100 each in quantities of 1 to 10. Volume discounts also are available.

Support. All registered users may telephone MDBS and leave a message for a call back from technical support. KnowledgeMan Priority Support, designed especially for applications developers, is available for $300 a year and allows four hours of telephone support. In addition, MDBS offers two books on KnowledgeMan and sponsors seminars in major cities.

—*Julie Anderson*

It is easy to forget at this point that items have been rearranged to facilitate data entry from screens. Automatic checking to head off duplicate entries can be accomplished using split forms. (Multiple forms can be presented on a screen simultaneously or sequentially.) For example, suppose the user wants to prevent duplicate entries in the author name field. Two forms are set up on the screen (it looks like one form to the operator). The only field in the first form is for Social Security Numbers. As soon as this entry is made, the program searches the Author table for a duplicate. If one is found, the operator is told that the author is already in the table, and a screen is presented to edit the existing author information. If no duplicate is found, data input continues with the second form in order to create a new author entry.

AD HOC QUERIES

One of the questions asked by the *PC Tech Journal* sample application is the average fee paid to authors. Answering this question is simple because it involves only one table and takes advantage of KMAN's automatic statistics computation feature. Whenever the environment variable E.STAT is TRUE, KMAN automatically computes statistics for all listings or selections. For example, the following command

```
LIST Title, Payment, Bonus, Payment +
  Bonus
```

does the job as long as the Article table is open and is also the default table. From left to right, the columns are Title, Payment, Bonus, and the sum of Payment plus Bonus for each article. At the end of the listing, KMAN automatically generates the count, sum, average, variation, standard deviation, minimum, and maximum of each numeric column (Payment, Bonus, and Bonus + Payment). If the user wants only the statistics and not a listing of articles, the command STAT instead of LIST would be adequate.

Consider this query: how much was paid per printed editorial page (including only articles that are not departments) in an issue?

```
LIST (Payment + Bonus) / (Edpage +
  Listpage)
    WHERE Category <> Department
    AND Volume = 3 AND Number = 2
```

This command generates the data by issue (Volume and Number identify an issue) and the statistics feature calculates the averages at the end of the listing. Again, using the command STAT instead of LIST generates only the average for the issue. Average per article within the issue is suppressed.

This next query asked by the sample application involves multiple tables: which articles were received after the deadline for an issue? To answer this query the user must go to the Issue file to find the deadline and to the Article file for the date submitted, then compare the date submitted to the deadline. Comparing dates requires the TOJUL function that converts dates from the current format, as specified by the E.DTYP variable, to the Julian equivalent. This query would be handled by the following command:

```
LIST Article.Title Article.Datein
  Issue.Deadline
    FROM Article FROM Issue
    WHERE TOJUL(Issue.Deadline) <
    TOJUL(Article.Datein)
```

Another way of handling dates is to create a virtual field in the table to assemble the parts of a date into a year-month-day string. This field can be used directly in selections and sorts.

One of the sample application tasks requires using multiple tables with many-to-many relations. A columnar report must be generated that lists for each issue the article titles, author

TABLE 1: *Benchmark Results*

BENCHMARK TASK	TIME (secs)
Add 900 records to an empty database table	206
Index table on two fields (7 bytes)	26
Document and tally codes from one field	
Using the STAT command	90
Executing a program to count values	64
Mass change of one field (28 rows of 900)	17
Extract selected records to create a text file	21

Compared with other data management systems reviewed in this series, KMAN seems to be average in many respects—a little quicker than most in indexing and a little slower than most in counting occurrences of record values.

names, number of editoral pages, and number of listing pages per article. The data per issue should be calculated for editorial page and listing page counts and the total given for all pages.

Two new fields can be created that facilitate the generation of this report. First, a virtual field, named Artisu, can be added to the Article table where Artisu = (Volume * 100) + Number. This is a shortcut so an issue can be described by a single field (Artisu) rather than two—Volume and Number. Life is also made easier by creating a virtual field for the author names:

```
LFname = trim(Lname)+", "+Fname.
```

LFname now can be used for sorting or direct output without further formatting as though it were a single field rather than a combination of two fields.

Basically, to solve the many-to-many problem, the program picks a logical starting place—in this case table Article.ITB— and obtains a record. If a new issue is involved, the subtotals for the previous issue (if one exists) must be printed. If not, the user can continue to find information about the current article. To do so, the AutArt table is consulted to match the existing article number with an author number (or more), and the author table is searched using the author number in order to retrieve author name. The authors are listed, edit and list page counters incremented, and the next article examined.

BENCHMARKS

The benchmark tests and results are shown in table 1. Compared with other data management systems reviewed in this series, KMAN seems to be average in most respects—a little faster than most in indexing and a little slower in the tasks that involved counting occurrences of record values.

The first benchmark, loading the author database, required 206 seconds. Benchmark 2 required the Author table to be indexed by state and zip code. The KMAN command for this is:

```
INDEX "Iauthor.IND" FOR Author BY AZ
  State AZ Zip
```

Using the index requires that an index file be named in the USE command along with the associated data table, as in the following example:

```
USE "Author.ITB" with "Iauthor.IND"
```

The program required only 26 seconds to handle the two-field index task.

KMAN can be quite time sensitive in some tasks. The serious program developer may want to try several approaches to a task if time is critical to the application. Benchmark 3 requires a count of occurrences of authors' home states. The easiest way to solve this problem is to use the STAT command after turning off all statistics except count. The end single line request looked like this:

```
STAT State
```

About 90 seconds later the task was complete. The same task was attempted by writing a program that reads the Author table one record at a time in state sequence, counting the states. Surprisingly, the entire task took only 64 seconds when run under the program, and the output was more attractive.

Benchmark 4, changing all CO state codes to CL, is accomplished with a one-line command:

```
CHANGE State TO "CL" FOR State = "CO"
  ALL.
```

The time required for this task was 17 seconds, about average for microcomputer data management systems.

Benchmark 5 called for extracting selected records—in this case, authors from California—and writing a report to a disk file. Again it could be handled by a single command from the KMAN prompt. Setting the environment variable E.ODSK to TRUE opens the disk file DSKOUT.TXT. The command to collect unique California authors is:

```
LIST Lname Fname State
    FOR State = "CA"
    ORDER BY AZ Zip.
```

BIG GUNS

KnowledgeMan is far from perfect, but its wealth of features lets the user forget the imperfections or, at least, work around them. MDBS has a fine reputation for designing excellent relational and postrelational data managers. This product measures up to MDBS's promises not only with developers but also in the end-user market.

KMAN is a reliable product. It had many bugs in its early days, but none that caused data loss. The product has no crash recovery utility.

In its latest version, KMAN has gotten on-line, context-sensitive help, roll-down menus, and optional utilities that integrate the database function with a text processor, a communications package, a report writing utility, a forms painter, and a mouse driver. Optional language interfaces are becoming available (C exists now in the optional K-C module) as are LAN, VAX, and XENIX/UNIX versions. With the existing CP/M and DOS configurations, KMAN will run on a variety of machines.

With all its bells and whistles, the KMAN system is expensive—more than $1,600 retail with all the available options. An upgrade from version 1.07 (the last 100-series version) to 2.0 is $295. Factory support is available at several different response levels for varying costs. Free support consists of a call-back system in which the user must call and leave a message, then wait for a support representative to call back—which may mean two or three days.

KnowledgeMan is a professional applications development system. Certainly the novice can use it for simple applications, but KMAN is really designed as big guns for big gun applications. Putting up with the minor bugs is worth the opportunity to work with this generally excellent software offering. KMAN is so rich with features that the question has never been "can KMAN do such and such," but rather "how do you get it to do such and such."

Richard N. Aarons is senior editor at Business and Commercial Aviation *magazine where he evaluates aviation-related software and computer systems. He conducts seminars across the country on the use of microcomputers in flight department management.*

THE PROGRAMMER'S SHOP

helps save time, money and cuts frustrations. Compare, evaluate, and find products.

RECENT DISCOVERY

PC Scheme LISP - by TI. SCHEME has special, simple, "orthogonal" syntax. Lexical scoping, block structure, call by value, and tail-recursive semantics. Compiler, EMACS-like editor, DOS. Supports debugging, graphics, windowing. PC $ 95

AI-Expert System Dev't

Arity System - incorporate w/C. PC $295
Experteach - Powerful, no limit on memory size. Samples PC $399
EXSYS - All RAM, Probability.
Why. Trees. Solid. files. popular PC $359
1st class - by example, interfaces $250
Insight 1 - Probabilities, required thresholds, menus, fast ($79)
Insight 2 adds backward, forward, partitions, dB2, lang., access. MS $399
Others: APES ($359), Advisor ($949), ES Construction ($100), ESP ($845), Expert Choice ($449)

AI-Lisp

List Our
GC LISP Interpreter - "Common", rich, Interactive Tutorial $ 495 Call
GC LISP 286 Developer - 2 to 15 meg RAM, compiler & Interp. $1195 Call
Microsoft MuLisp 85 $ 250 $199
TLC LISP - "LISP-Machine" - like, all RAM, classes, **compiler.** MS $225
TransLISP - Good for learning MS $ 75
WALTZ LISP - "FRANZ LISP" - like, big nums, debug, CPM-80 MS $149
Others: IQ LISP ($155), BYSO ($125),

AI-Prolog

ARITY Standard - full, 4 Meg
Interpreter - debug, C, ASM PC $ 350
COMPILER/Interpreter-EXE PC $ 795
With Exp Sys, Screen - KIT PC $1250
MicroProlog - enhanced MS $ 229
MProlog - Improved, Faster PC $ 475
Professional MicroProlog MS $ 359
Prolog-86 - Learn Fast MS $ 95
Others: Prolog-I ($365), Prolog-2 ($1795)

Free Literature Compare Products

Evaluate products. Compare competitors. Learn about new alternatives. One *free call* brings information on just about any programming need. Ask for any ***"Packet"*** or Addon Packet □ AI □ ADA, Modula □ BASIC □"C" □ COBOL □ Editors □ FORTH □ FORTRAN □ PASCAL □ UNIX/PC or □ Debuggers, Linkers.

Our Services:

- Programmer's Referral List
- Compare Products
- Help find a Publisher
- *Evaluation Literature FREE*
- BBS - 7 PM to 7 AM 617-826-4086
- Dealers Inquire
- Newsletter
- Rush Order
- Over 700 products
- National Accounts Center

C Language-Compilers

BDS C - solid value, fast CPM80 $125
C86 by CI - 8087, reliable MS $299
Consulair Mac C w/toolkit MAC $299
ECP C/88 MS $ 60
Lattice C - from Lifeboat MS $289
Lattice C - from Lattice MS $339
Mark Williams - w/debugger MS $399
Megamax - tight full **ATARI/ST $179**
Microsoft C 3.0 - new MS $259
Q/C 88 by Code Works - Compiler source, decent code, cross/native MS $125
Wizard C - Lattice C compatible, full sys. III, lint, fast. MS $379

C Language-Interpreters

C-terp by Gimpel - full K & R, .OBJ and ASM, large progs. MS $249
INSTANT C - Source debug, Edit to Run-3 seconds MS $399
Interactive C by IMPACC Associates. Interpreter, editor, source, debug. PC $225
Introducing C - learn C fast, self paced tutorial PC $109
Professional Run/C - Run/C plus create add-in libraries, load/unload them. MS $199
Run/C - improved MS $109

C Support-Systems

Basic C Library by C Source $139
dBrief, the dBASE Assistant PC $ 95
dBase Tools for C Tools $ 79
PC Graphics $ 79
C Sharp - well supported. Source, **realtime,** tasks $600
C ToolSet - DIFF, xref, source $135
Lattice Text Utilities $105
The HAMMER by OES Systems $179
SECURITY LIB - add encrypt to MSC. C86 programs. Source $250 PC $125

C Libraries-General

Application Programmer's Toolkit MS $375
Blaise C Tools 1 ($109), C Tools 2 $ 89
C Food by Lattice-ask for source MS $119
C Power Windows by Entelekon $119
C Utilities by Essential - Comprehensive screen graphics, strings. Source. PC $139
C Worthy Library - Complete, machine independent, source MS $295
Entelekon C Function Library PC $119
Entelekon Superfonts for C PC $ 45
Greenleaf Functions - portable, ASM $139
Polytron - for Lattice, ASM source $ 85
Software Horizons - Pack 1 PC $129

C Libraries-Communications

Asynch by Blaise $149
Greenleaf - full, fast $139
Software Horizons - pack 3 $119

C Libraries-Files

FILES: C Index by Trio - full B + Tree, vary length field, multi compiler
/File is object only $ 89
/Plus is full source $349
dbVISTA - full indexing, plus optional record types, pointers, Network.
Object only - MS C, LAT, C86 $179
Source - Single user MS $459
Source - **Multiuser** MS $929
CBTREE - multiuser record locking, sequential, source, no royalties MS $ 99

FEATURE

OBJECT-ORIENTED Programming: with High Performance METHODS Smalltalk Language

METHODS is a powerful, language-compatible subset of Smalltalk-80™ that is ideal for software prototyping, simulation, databases, advanced user interfaces, and numerous AI applications. Over 100 classes and 2000 methods — easily-customized software building-blocks for rapid, incremental development of real-world applications.

An open-ended window environment that encourages exploration while allowing recovery from any error. Extensively documented in a four-part manual that addresses the needs of everyone from beginners to experienced programmers.

Methods also offers a communications interface that includes remote UNIX™ access, an object-oriented shell for DOS programs, and much more.

AND NOW . . .

GRAPHIC METHODS — all of the above, converted to bit-mapped graphics, plus . . .

- Logo-style graphics using class "Pen"
- Multiple fonts
- Form editor, for freehand drawing with the mouse
- Bit editor, for the creation or modification of icons, fonts, and other graphic elements
- A new debugger with the ability to inspect and change variables and resume execution
- Supports CGA, EGA, Hercules, and AT&T Hi-Res graphic controllers. Available after June 1st.

METHODS PCDOS $239
GRAPHIC METHODS PCDOS $665

FEATURE

ASSEMBLER PROGRAMMERS: Be an Expert and Take Out the Pain with the First Assembler Interpreter/Debugger ADVANCED TRACE-86

A complete, interactive, visual Assembler Interpreter/Compiler is integrated with a powerful debugger and editor to create, debug, rewrite, and test code. Increase understanding while you watch on your screen all aspects of the main processor (8088, 8086, 80286) and floating point processor (8087, 80287).

Instructions from a full screen of disassembled code can be single-stepped or run at other speeds while you watch registers, flags, the stack, and memory. Set or suspend breakpoints by: address range, label, or conditions. Toggle to the user screen to see the effects.

Add or delete program statements or directly change memory and registers to recarve your program while debugging — don't lose your ideas by waiting to assemble.

Gone too far? You can even back up your program!

On-line help and demos help you learn quickly.

Directly access DOS functions or the 8087/287's math functions and quickly see exactly what results are produced.

If you spend even a few hours with assembler code, whether your own or from a compiler, you deserve to get frustrations out of your way with Advanced Trace-86.

Morgan Computing Co., Inc. **PCDOS $149**

We support MSDOS (not just compatibles), PCDOS, Xenix-86, CPM-80, Macintosh, Atari ST, and Amiga.

CIRCLE NO. 162 ON READER SERVICE CARD

THE PROGRAMMER'S SHOP

provides complete information, advice, guarantees and every product for Microcomputer Programming.

ORDER TODAY SPECIAL PRICE

C-TREE by Faircom - portable, stable, thorough, balanced Btree file manager with source. MSDOS List $395.
ORDER BEFORE 6/30/86 and mention this ad for a special price of $289.
ORDER AFTER 6/30/86 for $349.

RECENT DISCOVERY

Synergy - Create user interfaces. TopView-compatible multitasking operating environment offers windows, icons, pull-down menus and fonts in 12K RAM. MS $375

Editors for Programming

BRIEF Programmer's Editor - undo, windows, reconfigure	PC Call
C Screen with source	80/86 $ 75
EMACS by UniPress - powerful, multifile, MLISP. Source: $949	$299
Epsilon - like EMACS	PC $169
FirsTime by Spruce - Improve productivity. Syntax directed for Turbo ($69), Pascal ($229), or C ($239)	
Kedit - like XEDIT	PC $115
Lattice Screen Editor-multiwindow, multitasking Amiga $100	MS $125
PMATE - power, multitask	80/86 $159
VEDIT - well liked, macros, buffers CPM-80-86.	MS $119
XTC - multitasking	PC $ 85

MultiLanguage Support

BTRIEVE ISAM	MS $199
CODESIFTER - Execution PROFILER. Spot bottlenecks.	MS $109
Dan Bricklin's Demo Program	PC $ 75
HALO Graphics - Multiple video boards, printer, rich. Animation, engineering, business. Any MS language, Lattice, C86	PC $249
PLINK 86 - a program-independent overlay linker to 32 levels for all MS languages. C86 and Lattice.	MS $299
Pfinish Performance Analyzer by Phoenix	MS $299
Profiler by DWB Associates	MS $ 99
PS MAKE by UniPress	PC $129
Screen Sculptor - slick, thorough, fast, BASIC, PASCAL.	PC $109
ZAP Communications - VT 100, TEK 4010 emulation, file xfer.	PC $ 85

Fortran & Supporting

Forlib + by Alpha - graph, comm.	$ 59
Fortran >> C-FORTRIX C creates maintainable **translations.**	$995
MACFortran by Microsoft - full '77	$229
MS Fortran	$219
No Limit - Fortran Scientific	$129
PolyFortran - xref, pp, screen	$149
Prospero - '66, reentrant	$390
RM Fortran - enhanced "IBM Ftn"	$399
Scientific Subroutines - Matrix	$149
Statistician by Alpha	$269
Strings and Things - registers, shell	$ 59

Debuggers

Advanced Trace-86 by Morgan Modify code on fly.	PC $149
CSPRITE - data structures	PC $139
Periscope I - own 16K	PC $269
Periscope II - symbolic. "Reset Box." 2 Screen	PC $119
Software Source by Atron-Lattice, MS C, Pascal, Windows single step, 2 screen, log file.	MS $115
w/Breakswitch	$199
Pfix-86 Plus Symbolic Debugger by Phoenix - windows	PC $289

Atari ST & Amiga

We carry full lines of Manx, Lattice, Metacompco and Prospero.

Note: All prices subject to change without notice. Mention this ad. Some prices are specials. Ask about COD and POs. All formats available. UPS surface shipping add $3/item.

Xenix-86 & Supporting

Basic - by Microsoft	$279
Cobol - by Microsoft	$795
Fortran - by Microsoft	$399
PANEL Screen LIB - multi-language	$539
Xenix Complete Development System	$985

FEATURE

EXPERT SYSTEM DEVELOPMENT: Practical, Complete, and Unlimited Features Help Smoothly Build Expert Systems with EXSYS

EXSYS, Inc. has built a stable and complete toolkit by listening to users and examining what they need. One of the first Expert System Shells for the PC, EXSYS provides the features of just about all of its combined competitors, plus the documentation and examples you will need to learn in this field.

UNLIMITED FEATURES? EXSYS supports backward chaining of IF/THEN/ELSE rules, full math support, probabilities, explanations, the ability to call external programs including spreadsheets, database managers, or custom-written front ends with data passed to and from the external program, plus the ability to handle substantial applications of up to 5,000 rules. All user input is either English text, menu selection, or algebraic expression.

The systems developed can explain why information is needed and how it will be used. The ability to "change and rerun" allows expert system modelling of problems. Written entirely in C, EXSYS provides very high speed execution and efficient memory utilization.

But if these features are not enough, use the interface to Lotus, dBASE, BASIC, C, or any other .EXE or .COM file. Already in use at over 1,000 sites with many complex and powerful expert systems developed. Several expert systems have been marketed with a low-cost runtime license. Single computer use is only $349. Unlimited runtime distribution is available for an additional $539.

PCDOS $349

FEATURE

PROVE CONCEPTS, Tighten Designs, or Quickly Create Demos with DAN BRICKLIN'S DEMO PROGRAM

Focus exclusively on what the user sees and does — with complete, flexible tools for fast and tangible results.

Show simulations of keyboard input and screen content by using Dan's editor and screen building tools. "Snapshot" screens created by other programs, use a prior screen to create a new one. Draw boxes, cut and paste any screen subset, control attributes like inverse video and color. Use the graphics characters available in the text mode.

Cause screens to branch in any sequence using labels. Give the demo user control. Print-control helps put your slide show to paper or disk in a variety of formats.

PCDOS $75

FEATURE

PROGRAMMING TEAMS: Manage and Control Source Versions Efficiently with POLYTRON Version Control System (PVCS)

Duplicated efforts and time wasted rebuilding and cataloging source can be avoided. Reliably delegate control for source AND documentation to PVCS.

Save confusion and disk space by letting PVCS maintain the complete current version plus all of the increments, decrements, and related notes needed to rebuild any version. Maintains a complete history of changes. No experience with such a system is assumed. The documentation and examples will help you learn quickly. Privilege levels control access so the administrator knows who is working on each module. Now used in PC software development environments at Hewlett-Packard, Sperry, 3-COM, ROLM, and many more. Ask about discounts for LAN and multi-keyboard access.

POLYTRON **PCDOS $369**

Call for a catalog, literature, advice and service you can trust

NEW HOURS
8:30 AM - 8:00 PM EST

800-421-8006

THE PROGRAMMER'S SHOP™
128-P Rockland Street, Hanover, MA 02339
Mass: 800-442-8070 or 617-826-7531 486

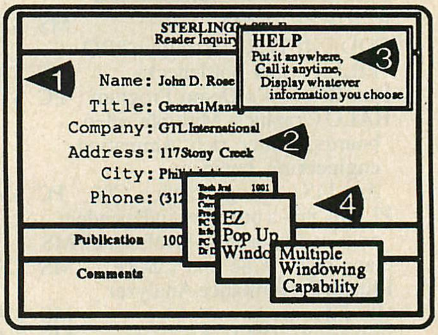
"I have been pleased with your catalog selection, the knowledge of your telephone answering staff, and the promptness of your service. I have every wish to become a continuing customer of the Programmer's Shop.

Carl C. Rollo

How to develop elegant software using Basic or "C"

by Cary Harwin

A widely shared dream is the fast, easy creation of error-free code and documentation. One of the secrets of top software designers is increased productivity through the use of powerful tools. Imagine the benefits of joining this group. Here are some ways.



For Software Designers Using BASIC :

BASIC Development Tools

1 **Screen Builder**++ easily creates powerful application screens. Using your favorite word processor or the built-in full screen editor you quickly paint your screen image. Simply create and edit fields of any length, anywhere on the screen, with both full screen editing within each field and movement from field to field. Enhanced graphics presentations are encouraged using all 255 ASCII characters. With available on-line help you are quickly up and running. Then error-free BASIC is automatically generated. Now the code is ready to insert into your application program without the use of BLOAD or BSAVE. With no BIOS calls, the code is tight and efficient. The system is complete with both a subroutine library and multiple examples. For only $59 you should be using it now.

2 **B-Plus Tree**++ is a powerful record handling subsystem. Write programs without worrying about sorting or handling of index files. Fast (access a key field in a file of 32,000 records in less than one second). Direct and sequential accessing, variable length keys, multiple non unique keys and multiple keys per record. Get **first**, **next**, **last** or **prior** key in sequence. Users are amazed at the source code's simple elegance and are delighted in the value pricing of $59.

3 **Help Message**++ systems are a part of every modern software application. Now you can provide your users with the help they expect. Interactive, context sensitive help reduces the need for extensive manual documentation while allowing your users to become quickly operational . High performance is assured through the inclusion of a fast assembler subroutine. Overhead to your program is insignificant as your text information is turned into callable libraries. A justifiable necessity at $49.

4 **EZ Screen**++ or Pop up Windows is an assembler routine embedded in a basic subroutine to perform screen handling and windowing functions when called from a BASIC program. Implementing EZ lets you create modern and fast menus, windows, notepads, etc. The power and flexibility found here rivals the windowing power of the modern languages found in the industry today and a must have at only $39.

Each of these stand alone but when used together they are truly outstanding. Prove it to yourself by trying all four.

A Pleasant Surprise - It is hard to find good, affordable software; so I've done something about it. For a limited time you can purchase all four BASIC Development Tools for only $89. ($206 value)

+Runs under interpretive BASICA or Microsoft Basic. Compilable under IBM, Microsoft, or QuickBASIC Compiler.

For Software Designers Using "C" :

BlackStar™ "C" Function Library*

The new generation of "C" compilers demand a full featured, high performance function library. With over 250 functions, BlackStar is "the" new standard of "C" function libraries. Created for internal development and used extensively for the past two years, BlackStar has only recently been released to outside developers. Now there is no need for you to re-invent the wheel. I believe that it has everything (&more) on your wish-list of functions.

Standard Functions:
Screen handling, strings, graphics, file I/O, directory searches, peripheral control, and extensive date/time
Supports device drivers
Utilities with source: document extract, pretty printer, greplike

Powerful New Enhancements:
Menu manager and Text windows
Microsoft Mouse support
Encryption
Table driven printer control
Table driven keyboard functions for complete keyboard control and customization

High Performance features:
Optimized (in assembler) for speed and compactness
Interrupt redirection.

Complete
Compatible with Lattice and Microsoft Compilers (Ver.3) - adaptable to others
Complete source code, demo programs and comprehensive users manual

If you are using another function library it is time to make this major upgrade.

Wow! - An outstanding value at our regular price of $99 but a virtual steal at our limited time special of only $79.

Amazing Offer - I promise your programming productivity will be enhanced if you use these tools. In addition, I am going to make testing them in your home or office so easy that it will be a shame if you don't take advantage of my offer. Simply order by calling our toll-free hotline. After you receive your tools, put them to work developing those challenging projects. Try them for 75 days. If you're not convinced that they are a fantastic value, simply return them and I'll refund your full purchase price. But act now and join that select group of top software designers.

ORDER HOTLINE
1-800/7-Castle
(in CA call: 213/306-3020)

Visa, MC, COD, and checks accepted. Shipping included in U.S. and Canada. Others please add $15. CA residents add 6% tax.

*Requires PC or MS Dos.

Trademarks: BlackStar (Sterling Castle), MS-DOS (Microsoft), IBM and PC-DOS (International Business Machines), Lattice (Lattice, Inc)

STERLING CASTLE
Helping You to Solve Problems

702 Washington Street Suite 174
Marina del Rey, CA 90292

P.S. Please remember that our products are not copy protected and we charge no royalties.

CIRCLE NO. 183 ON READER SERVICE CARD

Accessing the Print Queue

The print spooler functions of DOS 3.x save on program development time and present a standard interface to the end user.

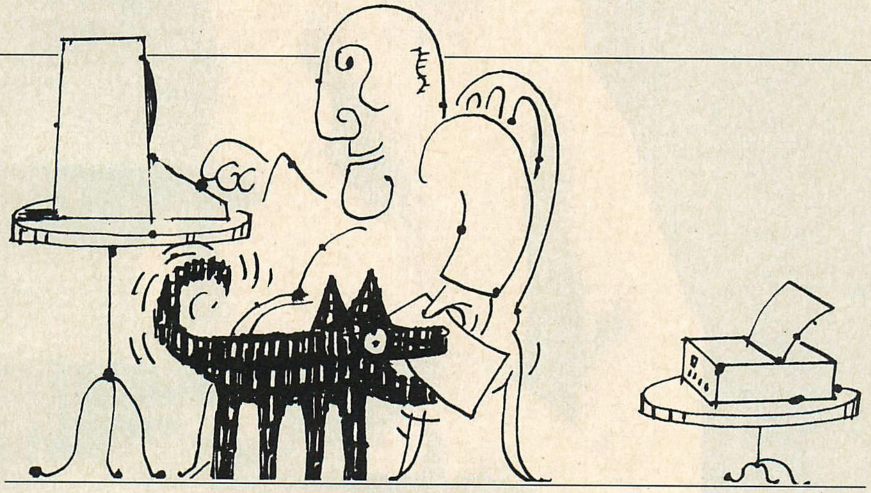


ILLUSTRATION • MACIEK ALBRECHT

Print spoolers and print buffers allow users to move on to other applications without waiting for the computer to finish printing a spreadsheet or report. In many cases, print buffers make it possible for users to operate without having to purchase a new, faster, and more expensive printer.

A hardware buffer receives and stores output as fast as the computer can generate it, then sends the output to the printer at a slower rate. Thus, the computer thinks it is finished printing long before the printer has received the last word. This approach can be an effective strategy as long as the buffer is equipped with an adequate amount of memory. Hardware buffers, however, are not always the best solution because they add to desktop clutter.

A software buffer performs the same function as its hardware counterpart by reserving a block of system memory to be used as a buffer. Periodically, the operating system executes a program called a print spooler to transfer characters from the buffer to the printer. This approach also can be effective but steals valuable memory space from spreadsheets, word processing programs, and other applications.

Several software vendors have recognized the user's need for access to background printing and have incorporated special background printing capabilities into their products. The basic drawback to this approach is that the background spooler terminates when the application is ended, leaving any files waiting to be printed in limbo until that application is started again.

For a print spooler to work across applications, it must be part of the PC's operating system. IBM recognized this and began providing pieces of a background spooler in DOS 2.0 with the PRINT.COM command.

The first time PRINT is executed, it is loaded into memory and becomes a permanent part of the operating system. Subsequent executions of PRINT can be used to add and delete file names from the print queue. Periodically, DOS interrupts whatever program is running and passes control to the spooler. The spooler is allowed a fixed amount of time to transfer data from the first file listed in the print queue to the output device. Control of the system is then returned to the application.

Unfortunately, the DOS 2.0 PRINT command allows access to the print queue only through a DOS command; users still cannot take advantage of the background spooler's capabilities from within an application.

With DOS 3.0 came several new features that made the background print spooler more useful. A library of spooler routines presented later in this article help users gain easy access to the DOS 3.0 background print spooler.

DOS 3.0 has an expanded PRINT command that provides better user control over the operating parameters of the background spooler. BASIC 3.0 also features a SHELL command that enables interpretive BASIC users to execute any .COM, .EXE, or .BAT routine from within a BASIC program. In other words, users can start the background spooler and submit files for printing from within BASIC simply by issuing the SHELL command and specifying DOS PRINT as its parameter. For example, to start the spooler and direct its output to LPT1:, the user types the command:

```
SHELL "PRINT /D:LPT1"
```

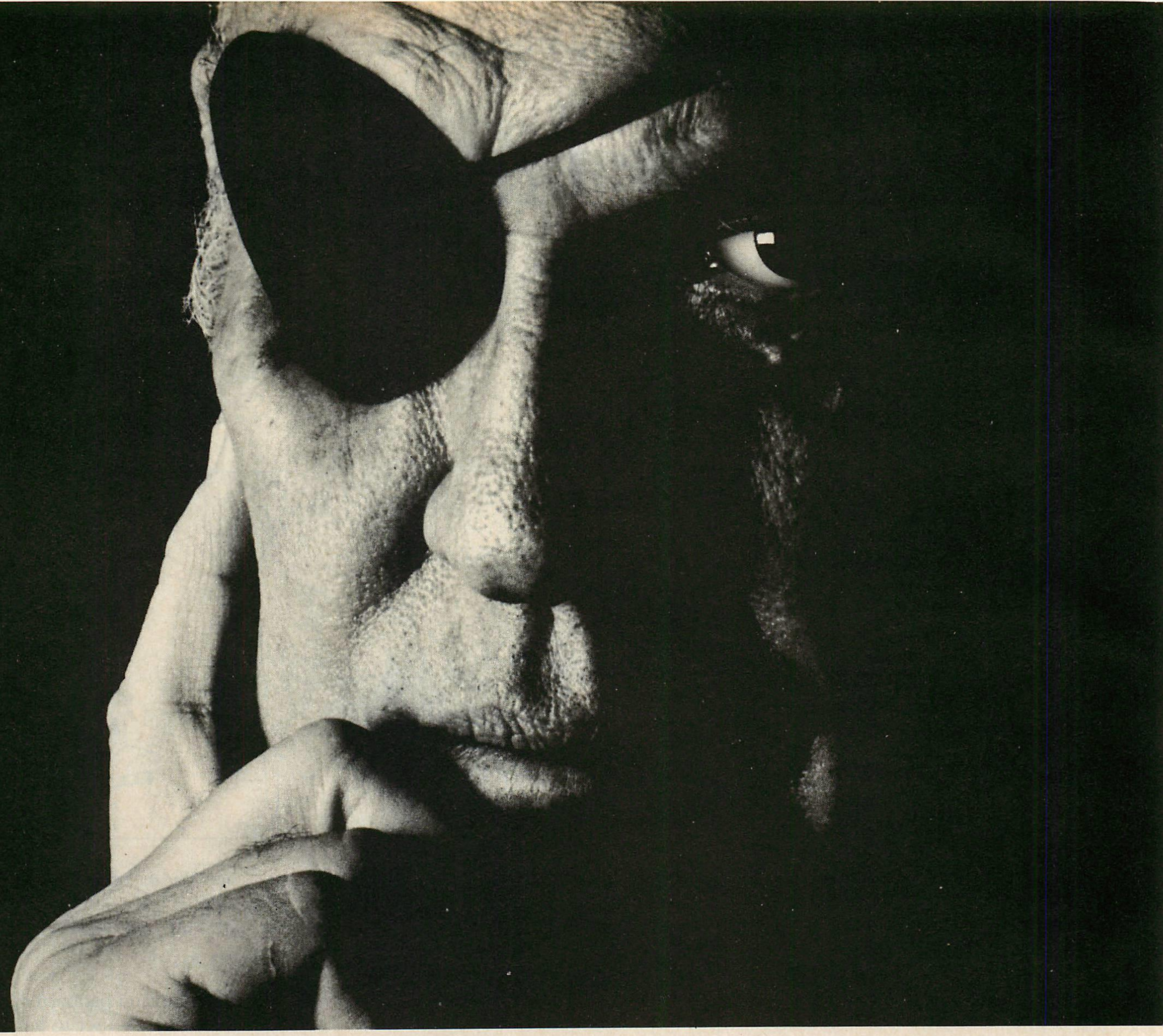
Other PRINT command parameters also can be included by the user to help establish the desired print spooler environment. These parameters can specify the size of the internal buffer, the maximum number of files in the queue, and the amount of time the operating system should allow for the spooler to transfer output to the printer.

After the background spooler has been started, the user can submit files for printing by issuing the command:

```
SHELL "PRINT d:\path\filename.ext"
```

When using the PRINT command from DOS or BASIC, the name of the file being referenced should be fully qualified by specifying both the drive and path. The wildcard characters * and ? also can be used to queue multiple files for printing by the background spooler.

The most important addition to DOS 3.0 was a set of interrupt service routines that provide programmers writing in languages other than interpretive BASIC with full access to the DOS background print queue from their applications. In all, six functions are available to the programmer via



Software pirates ruined our reputation

That was just for starters. I'm still paying fines and legal fees because a few employees thought they were saving me money. And it would have been so easy to avoid.

Software piracy is a Federal crime. At ADAPSO, the computer software and services industry association, we're doing everything we can to keep business leaders like you informed of the problem.

You can preserve the integrity of your company and protect it from liability by actively discouraging software piracy. ADAPSO can provide you with materials that can help, such as sample policy statements and information booklets for your employees.

Return the coupon to receive a full packet of information. It's free. And it could very easily keep your company off the hook.

Send this coupon or your business card for more information. Or call us at (703) 522-5055 and ask for Muriel.
ADAPSO, 1300 North Seventeenth Street, Arlington, Virginia 22209

NAME TITLE

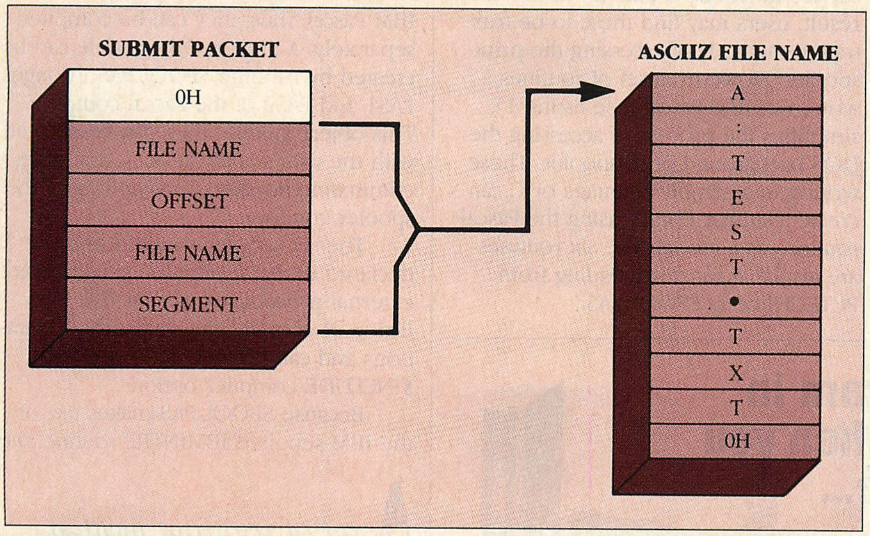
COMPANY

ADDRESS

CITY STATE ZIP

ADAPSO Software Piracy is a Federal Crime

FIGURE 1: *File Names*



The above structure specifies the file a:test.txt for print spooler commands. A submit packet is a five-byte structure that points to the file name in ASCIIZ format. The full file name string is not shown; these ASCIIZ strings are 64 bytes long.

DOS interrupt 47 (2FH) that can be used to control the queue. In each case, the user must set the AL register equal to the desired function number and the AH register equal to 1 before executing interrupt 2FH. Some of the function calls have additional input parameters, which are described below.

After issuing the particular function interrupt, the user can check for successful execution by testing the carry bit of the flags register. If the carry bit is not set, the user is assured that the function executed successfully and any return values set in the registers are valid. However, if the carry bit is set, an error was encountered and the AX register contains a DOS error code.

Function 0 can be called to determine if the DOS background print spooler has been installed and is available for use. A return code of 255 in the AL register indicates the spooler is ready to receive the names of files to print. A return code of 0 means the spooler has not yet been installed using the DOS PRINT command. A value of 1 indicates not only that the spooler is not available, but also that it cannot be installed either because the system is not running under DOS 3.0 or because interrupt 2FH already is being used for some other purpose.

Function 1 adds a file name to the DOS background print spooler queue. Before executing the interrupt, the register pair DS:DX must point to the segmented address (segment:offset) of a five-byte structure called a *submit packet* (see figure 1). The first byte of this packet must be a hex 0. Bytes 2 through 5 are set equal to the segmented address of the ASCIIZ string that contains the drive, path, file name, and extension of the file to be added to the print queue. An ASCIIZ string is a 64-byte structure that contains the desired data and is terminated by an 0H of length 1 byte. When specifying the segmented address of the ASCIIZ string in the submit packet, bytes 2 and 3 must point to the offset portion, and bytes 4 and 5 to the segment portion.

Function 2 allows the user to delete a file name from the background print queue. Before calling this function, the user must set the register pair DS:DX to the segmented address of an ASCIIZ string which contains the drive, path, and file name of the file to be deleted from the queue. The wildcard characters * and ? are valid in function 2 and can be used to delete a range of files from the queue with a single call.

The entire print queue, including the file currently printing, can be cleared with function 3. This function has no additional input parameters.

Function 4 suspends the spooler and returns the segmented address of the print spooler queue in the register pair DS:SI. The spooler queue is a 2,048-byte block of memory, which is large enough to hold 32 spooler entries. Each entry is a 64-byte ASCIIZ string. If a spooler entry is shorter than 64 bytes, it is terminated by a byte of 0, and all characters from that position out to the 64th byte are meaningless. The end of the queue is marked by a 0 in

the first byte of an entry. Users should not try to shortcut the spooler by modifying the print queue directly; the service functions always should be used to ensure reliability.

After function 4 has been executed, function 5, which has no additional input parameters, can be used to unlock the queue and resume normal print spooler operations.

Dealing with segmented addresses, ASCIIZ strings, and submit packets can get a bit tricky, and complete and accurate documentation of the 2FH interrupt service functions is difficult to find. As a result, users may find these to be frustrating methods of accessing the print spooler. However, a set of routines, written in IBM Pascal (see listing 1), simplifies the process of accessing the DOS background print spooler. Those writing in assembly language or C can create a similar library using the Pascal routines as a model. The six routines are available for downloading from PCTECHline as SPOOL.PAS.

This library of spooler routines takes advantage of the module facility of IBM Pascal; thus, they can be compiled separately. A SPOOL.OBJ module can be created by running SPOOL.PAS through PAS1 and PAS2 of the Pascal compiler. This object module then can be merged with the user's application at link time, eliminating needless recompiling of the spooler routines.

The six spooler routines must be declared in the application program as external procedures. SPOOL.INC (see listing 2) contains the necessary declarations and can be included using the $INCLUDE compiler option.

Because SPOOL.PAS makes use of the IBM-supplied IBMINTRP routine, the IBMPAS.LIB library must be specified by the user in addition to the PASCAL.LIB library when the application program and the spooler procedures are linked. If the application program is called MAIN.PAS, the user should specify **main + spool** as the object modules and **ibmpas + pascal** as the libraries.

A set of routines written in IBM Pascal simplifies the process of accessing the DOS background print spooler.

Each procedure in SPOOL.PAS includes a list of values for the return code, which should be checked for errors. The GET_SPOOL_STATUS return codes have been changed from the error codes described in function 0 to give RETCODE a more consistent meaning across the spooler procedures.

The spooler routines should be used only with DOS 3.0 or later. An additional routine included in the library (GET_DOS_VERSION) uses the standard DOS function call to determine the version of the operating system.

The printer cannot be used for any other purpose while files are waiting in the queue. If the user accesses the printer directly before all files listed in the queue have been printed (by, for example, pressing Shift-PrtSc to print the screen), an "out-of-paper" or "printer time-out" error results.

Patrick J. Finan is a senior systems analyst with a degree in electrical engineering.

LISTING 1: SPOOL.PAS

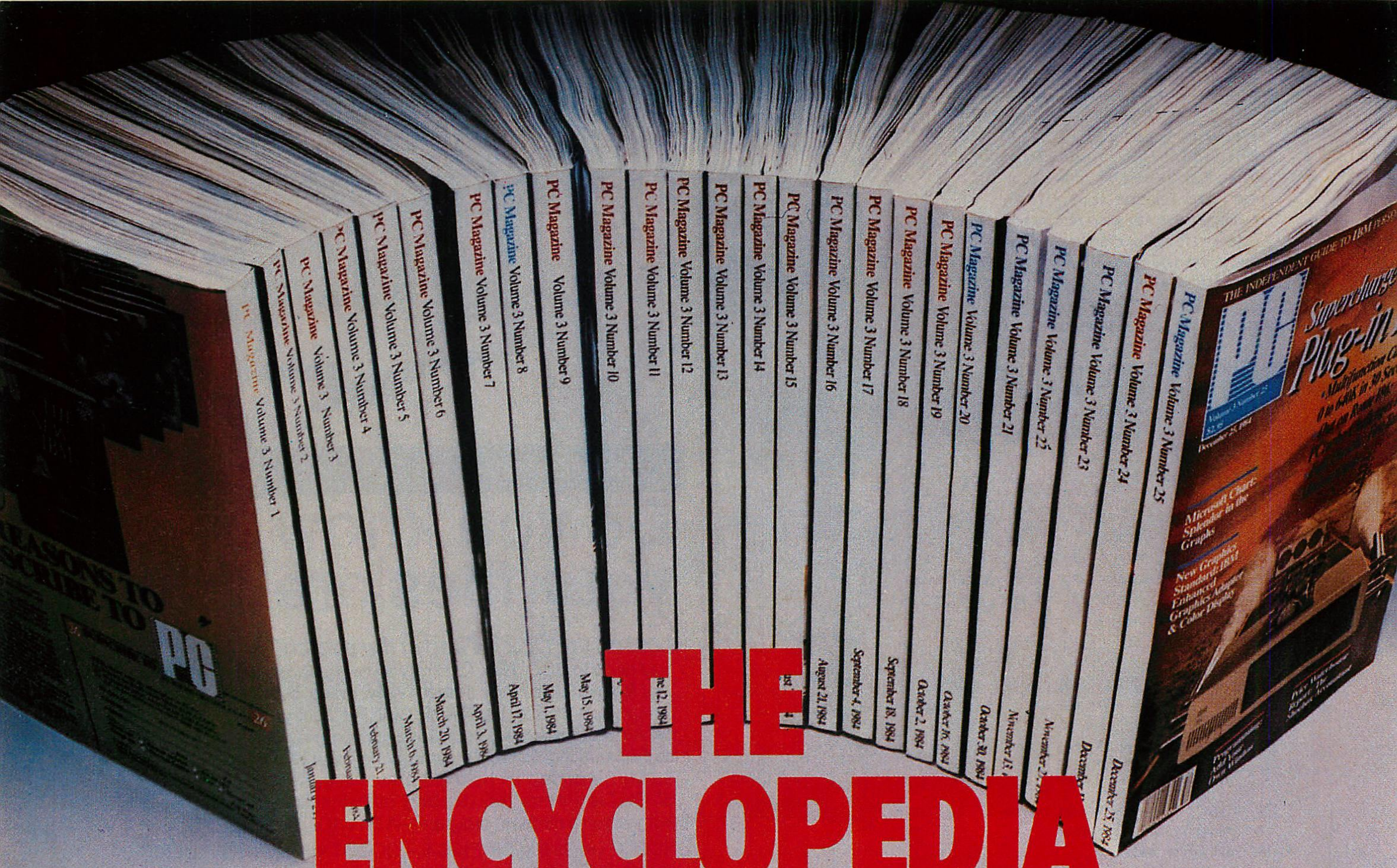
```
{           IBM PASCAL LIBRARY              }
{              SPOOL.PAS                    }
{         Background Print Spooler          }
{             for the IBM PC                }
{Copyright (c) 1986 Patrick J. Finan        }
{$INCLUDE: 'IBMINTRP.INT'}
MODULE SPOOL [PUBLIC];
   USES IBMINTRP;

CONST
   ERROR_MASK = 16#0001;

TYPE
   SUBMIT_PACKET = RECORD
      LEVEL_CODE     [00]: BYTE;
      FILE_OFFSET    [01]: WORD;
      FILE_SEGMENT   [03]: WORD;
      END;

   ASCIIZ         = LSTRING(64);
   BLOCK          = ARRAY[1..2048] OF CHAR;

PROCEDURE GET_DOS_VERSION(VAR MAJOR   : BYTE;
                          VAR MINOR   : BYTE;
                          VAR RETCODE : INTEGER);

{ MAJOR = 3 and MINOR = 10 when running under   }
{ DOS 3.10.                                     }
{                                               }
{ The values for RETCODE are as follows:        }
{   0 - DOS version in MAJOR and MINOR.         }
{  >0 - Error encountered... MAJOR AND MINOR may }
{       not be valid.                           }

VAR
   REGISTER    : REGLIST;

BEGIN {GET_DOS_VERSION}
   REGISTER.AX := 16#3000;
   INTRP(16#21,REGISTER,REGISTER);
   IF (REGISTER.FLAGS AND ERROR_MASK) = ERROR_MASK
      THEN RETCODE := ORD(REGISTER.AX)
      ELSE RETCODE := 0;
   MAJOR := LOBYTE(REGISTER.AX);
   MINOR := HIBYTE(REGISTER.AX);
END; {GET_DOS_VERSION}

PROCEDURE GET_SPOOL_STATUS(VAR RETCODE : INTEGER);

{ The values returned in RETCODE are as follows: }
{  -3 - Error in status request.                }
{  -2 - Spooler not available.                  }
{  -1 - Spooler not installed.                  }
{   0 - Spooler installed and available.        }
{  >0 - DOS error code.                         }

VAR
   REGISTER    : REGLIST;

BEGIN {GET_SPOOL_STATUS}
   REGISTER.AX := 16#0100;
   INTRP(16#2F,REGISTER,REGISTER);
   IF (REGISTER.FLAGS AND ERROR_MASK) = ERROR_MASK
      THEN RETCODE := ORD(REGISTER.AX)
      ELSE CASE (REGISTER.AX AND 16#00FF) OF
                0: RETCODE := -1;
                1: RETCODE := -2;
              255: RETCODE :=  0;
         OTHERWISE RETCODE := -3;
           END;
END; {GET_SPOOL_STATUS}

PROCEDURE SUBMIT_TO_SPOOL(VAR FILENAME : ASCIIZ;
                          VAR RETCODE  : INTEGER);

{ FILENAME := 'd:\path\path\filename.ext'       }
{                                               }
```

```
{ The values for RETCODE are as follows:          }
{   0 - File successfully submitted.              }
{  >0 - DOS error code.                           }

VAR
   PACKET   : SUBMIT_PACKET;
   REGISTER : REGLIST;

BEGIN {SUBMIT_TO_SPOOL}
   CONCAT(FILENAME,CHR(0));
   PACKET.LEVEL_CODE   := 16#00;
   PACKET.FILE_SEGMENT := (ADS FILENAME[1]).S;
   PACKET.FILE_OFFSET  := (ADS FILENAME[1]).R;
   REGISTER.AX  := 16#0101;
   REGISTER.DS  := (ADS PACKET).S;
   REGISTER.DX := (ADS PACKET).R;
   INTRP(16#2F,REGISTER,REGISTER);
   IF (REGISTER.FLAGS AND ERROR_MASK) = ERROR_MASK
      THEN RETCODE := ORD(REGISTER.AX)
      ELSE RETCODE := 0;
END; {SUBMIT_TO_SPOOL}

PROCEDURE DELETE_FROM_SPOOL(VAR FILENAME : ASCIIZ;
                            VAR RETCODE  : INTEGER);

{ FILENAME := 'd:\path\path\filename.ext'         }
{ The wildcard characters * and ? are valid.      }
{                                                 }
{ The values for RETCODE are as follows:          }
{   0 - file successfully deleted.                }
{  >0 - DOS error code.                           }

VAR
   REGISTER : REGLIST;

BEGIN {DELETE_FROM_SPOOL}
   CONCAT(FILENAME,CHR(0));
```

```
    REGISTER.AX := 16#0102;
    REGISTER.DS  := (ADS FILENAME[1]).S;
    REGISTER.DX := (ADS FILENAME[1]).R;
    INTRP(16#2F,REGISTER,REGISTER);
    IF (REGISTER.FLAGS AND ERROR_MASK) = ERROR_MASK
      THEN RETCODE := ORD(REGISTER.AX)
      ELSE RETCODE := 0;
END; {DELETE_FROM_SPOOL}

PROCEDURE CLEAR_SPOOL(VAR RETCODE : INTEGER);

{ The values for RETCODE are as follows:          }
{    0 - spool cleared.                           }
{  >0 - DOS error code.                           }

VAR
    REGISTER : REGLIST;

BEGIN {CLEAR_SPOOL}
    REGISTER.AX := 16#0103;
    INTRP(16#2F,REGISTER,REGISTER);
    IF (REGISTER.FLAGS AND ERROR_MASK) = ERROR_MASK
      THEN RETCODE := ORD(REGISTER.AX)
      ELSE RETCODE := 0;
END; {CLEAR_SPOOL}

PROCEDURE GET_SPOOL_QUE(VAR QUE_BUFFER : BLOCK;
                        VAR RETCODE    : INTEGER);

{ This routine will return the current DOS print }
{ queue in QUE_BUFFER.                           }
{                                                }
{ The values for RETCODE are as follows:         }
{    0 - request was successful.                 }
{  >0 - DOS error code.                          }

VAR
    REGISTER      : REGLIST;
    DOS_BUFFER    : ADS OF BLOCK;

BEGIN {GET_SPOOL_QUE}
    REGISTER.AX := 16#0104;
    INTRP(16#2F,REGISTER,REGISTER);
    IF (REGISTER.FLAGS AND ERROR_MASK) = ERROR_MASK
      THEN RETCODE := ORD(REGISTER.AX)
      ELSE RETCODE := 0;
    DOS_BUFFER.S := REGISTER.DS;
    DOS_BUFFER.R := REGISTER.SI;
    MOVESR(DOS_BUFFER, ADS QUE_BUFFER[1], 2048);
    REGISTER.AX := 16#0105;
    INTRP(16#2F,REGISTER,REGISTER);
END; {GET_SPOOL_QUE}

END.
```

LISTING 2: SPOOL.INC

```
TYPE
    ASCIIZ = LSTRING(64);
    BLOCK  = ARRAY[1..2048] OF CHAR;

PROCEDURE GET_DOS_VERSION(VAR MAJOR   : BYTE;
                          VAR MINOR   : BYTE;
                          VAR RETCODE : INTEGER); EXTERN;

PROCEDURE GET_SPOOL_STATUS(VAR RETCODE : INTEGER); EXTERN;

PROCEDURE SUBMIT_TO_SPOOL(VAR FILENAME : ASCIIZ;
                          VAR RETCODE  : INTEGER); EXTERN;

PROCEDURE DELETE_FROM_SPOOL(VAR FILENAME : ASCIIZ;
                            VAR RETCODE  : INTEGER); EXTERN;

PROCEDURE CLEAR_SPOOL(VAR RETCODE : INTEGER); EXTERN;

PROCEDURE GET_SPOOL_QUE(VAR QUE_BUFFER : BLOCK;
                        VAR RETCODE    : INTEGER); EXTERN;
```



THE ENCYCLOPEDIA OF INFORMATION FOR YOUR IBM PC

Taken one issue at a time, *PC* is the best source of up-to-date news and information about the IBM PC you can find anywhere.

But collect a few, and suddenly you realize that you've assembled a comprehensive reference work covering every significant new IBM PC application, product and development. *An encyclopedia you receive 22 times a year!*

Each volume of *PC* contributes the latest information on a wide variety of subjects including:

- **New Products** What's the latest in printers, word processing software and database managers? And what's best for your needs? Everything you need to know is in *PC*'s special reports and product reviews.
- **New Applications** Need to design a graphics presentation? Communicate with other PCs? Access a database? Find out how in *PC*'s expertly written articles.
- **New Developments** How are successful professionals and companies using their PCs? Find out in *PC*'s "Pro Columns" and use their proven techniques to move ahead in your own career.

Subscribe to PC today. Build your own IBM PC reference library by volume and save up to 50% off the regular one-year subscription price!

- One year (22 issues) only $21.97. (Save 37%).
- Two years (44 issues) only $34.97 (Save 50%).

To order, complete attached order card or call, toll-free,

1-800-852-5200

8ZC67

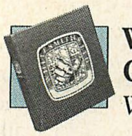
PC MAGAZINE

Reviews and Updates

ZIM RELEASE 2.5
Zanthe Information, Inc.

TAKETWO VERSION 1.0
United Software Security

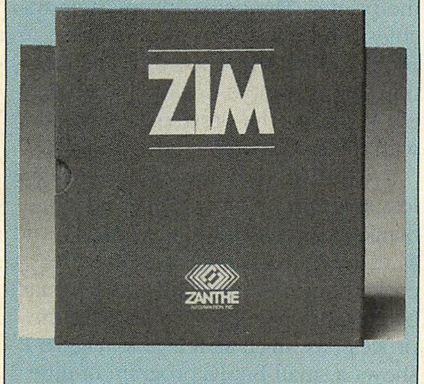
BASIC DEVELOPMENT SYSTEM
BetaTool Systems



WHITESMITH'S C COMPILER 3.01
Whitesmith's Limited

ZIM RELEASE 2.5
Zanthe Information, Inc.
1200-38 Antares Drive,
Nepean, ON K2E 7V2
613/727-1397

PRICE: $795



CIRCLE 349 ON READER SERVICE CARD

Release 2.5 of the ZIM data management system from Zanthe Information includes significant extensions to ZIM's forms management, database organization, and set manipulation capabilities, as well as a new user's manual.

The most visible extension to ZIM's software is the new *displays* facility for data entry forms management. Previous versions of ZIM allowed application programs to paint multiple (nonoverlapping) forms on the screen for simultaneous viewing. However, only with brute force could a ZIM programmer treat the forms as a group. Now, under version 2.5, as many as 30 separately developed forms can be composed into a single display, which is treated as a unit by ZIM programs and by the ZIM forms manager.

The real power of the displays facility lies in its ability to show multiple instances of the same form, arranged either vertically or horizontally across the screen. By replicating a form within a single-screen display, a business application programmer can paint screens quickly, showing, for example, line items for all the parts in a manufactured assembly's bill of materials or all the invoices outstanding to a particular customer. Replicated forms are referenced from commands and application programs using a simple syntax that is similar to the one used in most computer programming languages for referencing array elements.

Version 2.5 also introduces an *areas* facility that allows a ZIM application to straddle multiple DOS directories or even multiple disk drives. Using areas, parts of a database can be distributed onto removable volumes, such as cartridge disks or diskettes, or onto RAM disks. For example, compiled procedures can be stored on RAM disks for faster access, and data files can be stored in a directory on a hard disk. A simple CONFIG.SYS-like file directs ZIM to the different locations of the various database components.

ZIM embodies the entity-relationship (E-R) data model. Much of its data management utility stems from its ability to use schema-encoded knowledge of logical interfile relationships to guide the construction of aggregations of data called *sets*. ZIM's set-building commands now include the **keep** operator, which performs the relational project operation by discarding unwanted columns and duplicate row values as a set is built. By eliminating superfluous data during set construction, **keep** speeds the process and reduces storage requirements for result sets. A new **compute** verb also speeds set operations by allowing commands to cull summary statistics from sets without actually constructing the set first.

ZIM's ability to deal cleanly with absent data distinguishes it from most other data managers and programming languages on the market. Most systems require that a special value, such as 0 or the empty string, be reserved by application programs and used to represent missing data. ZIM, however, allows data fields and variables to take the value **$null** (undefined). Any arithmetic expression involving a **$null** value evaluates to **$null**; any comparison involving **$null**, such as **10 >= $null** or **10 < $null**, is false.

ZIM expression syntax has been extended in release 2.5 to allow programmers and interactive users to take advantage of the product's handling of undefined values. The term **<expr> where <logical expr>** evaluates to **<expr>** if **<logical expr>** is true and to **$null** if **<logical expr>** is false. Case expressions of the form {**<expr1>**, **<expr2>**,...**<exprN>**] evaluate to the value of the first non-**$null** expression in the list. Case expressions consisting of lists of **where** expressions provide LISP-like flexibility in expression evaluation. Because expressions can appear in reports, commands, forms, and programs, the increased power of expressions extends throughout ZIM.

Expression syntax also has been extended to allow the use of embedded assignments, as is possible in C. This capability can be used to accomplish complex operations compactly. In ZIM, embedded assignments are most useful in nonprocedural contexts, such as user commands or report specifications.

The new manual is as precise and succinct as was its predecessor, but it treats novices to the program more gently. The improved organization and clear tutorials make ZIM more accessible to first-time users who may lack an extensive computer science background. In addition, amenities, such as section dividers and boldface page and topic headings, help to speed quick-reference use of the manual.

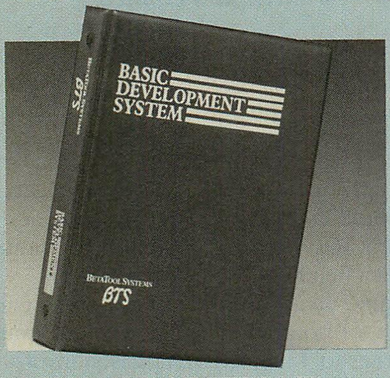
ZIM 2.5 is available free to users of version 2.4 licensed under Zanthe's software subscription program. Other 2.4 users can upgrade for $100.

—RICHARD M. FOARD

BASIC DEVELOPMENT SYSTEM

BetaTool Systems
8972 East Hampden Avenue,
Suite 179, Denver, CO 80231
303/793-0145

PRICE: $125



CIRCLE 351 ON READER SERVICE CARD

Much of the pain users experience when trying to get a big program off the ground now can be alleviated with the BASIC Development System (BDS). This product adds to the programming and debugging aids offered by the IBM BASIC system.

BDS is comprised of seven parts. XBASIC consists of a collection of miscellaneous helps, including command abbreviations, such as E (EDIT), L (LIST), P (LLIST), and D (DELETE). The U (Un-NEW) command undoes the work of NEW, provided the user has not exited to DOS with the SYSTEM command. Even more convenient are the several scrolling commands. These allow the listing of a program either by pages or a line at the time, both backwards and forwards. For example, Ctrl-PgUp lists the first page of a program on the screen; repeated PgDns list successive pages of the program.

SYSTEM TRON [LINE] [line number] produces a single-step trace of a BASIC program. **Line number** is an optional parameter that directs single stepping to begin at that location. If LINE is included, the line number of TRON is expanded to include the actual instruction. Line numbers and entire lines are displayed at the top of the screen; instruction display mode is toggled off and on with the gray minus key.

The utilities of BDS also include XREF, which lists all constants, line numbers, and variables along with the line numbers of their occurrences in the program. If a variable is modified in the instruction on a particular line, the line number is listed with an asterisk. Other parameters show, for example, the number of occurrences of the name of the variable in a particular line. XREF is most useful because it lists misspelled variables as well as those that are correctly spelled, and it saves the user from line-by-line proofreading for typographical errors.

FIND scans a BASIC program and produces a listing of strings or BASIC key words. F. **REM**, for example, lists lines containing remarks, even if an apostrophe is used instead of REM to denote remark lines. F, **REM** lists its findings on the printer. F. sends BASIC key-word references, such as FOR, GOTO, +, =, and >, to the screen.

VARIABLE DUMP, another utility included with BDS, is very helpful in debugging and can be used along with single stepping. The current values of all variables can be listed on the screen with V., while V, sends the output to the printer. The names of specific variables can be included as parameters in order to limit the amount of information that is printed or displayed. A simple V repeats the previous command with the same parameters.

COMPRESS allows BASIC to occupy less space in memory. Unnecessary blanks are removed in the process, and as many instructions as possible are packed onto a single line and separated by colons. UNCOMPRESS reverses the process, as much as it is able. Figure 1 shows a small BASIC program of nine lines. With COMPRESS (H is the operator symbol), this program can be squeezed onto two lines, and the byte count can be reduced by nearly half. UNCOMPRESS (I is the operator) can restore the program to the original expanded form (with a few more bytes because of the insertion of variable type definition symbols and the use of indenting). Compression can save 30 to 40 percent in storage space and 7 to 10 percent in running time.

Testing the VARIABLE DUMP utility using the V. operator on the expanded test program after execution results in the following output:

```
V.

126
A  !  108
I  %  9
J  %  10
```

Three variables had been defined in the test program: *A*, *I*, and *J*. *A* is a single-precision variable with a current value of 108. *I* and *J* are integers with current values of 9 and 10, respectively.

FIGURE 1: *Compression*

```
L
100 KEY OFF : CLS
110 DEFINT I, J
120 A = 0
130 FOR I = 1 TO 8
140 FOR J = 1 TO 9
150 A = A + 1.5
160 NEXT J
170 NEXT I
180 IF I < 5 THEN 120
OK
H

9 LINES, 119 BYTES IN OLD PROGRAM TEXT
2 LINES, 63 BYTES IN NEW PROGRAM TEXT
OK
L
100 KEY OFF : CLS : DEFINT I, J
120 A=0:FOR I=1 TO 8:FOR J=1 TO 9:A=A+1.5:
     NEXT J:NEXT I:IF 1<5 THEN 120
OK

I

2 LINES, 63 BYTES IN OLD PROGRAM TEXT
10 LINES, 135 BYTES IN NEW PROGRAM TEXT
OK
L
100 KEY OFF
101 CLS
102 DEFINT I, J
120 A! = 0
121 FOR I% = 1 TO 8
122   FOR J% = 1 TO 9
123     A! = A! + 1.5
124   NEXT J%
125 NEXT I%
126 IF I% < 5 THEN 120
```

The BDS compresses a BASIC program listing very efficiently—the sample file is reduced from 119 bytes to 63 bytes. The re-expansion helps to provide a clear listing.

An example of the cross-reference feature follows:

X.

0	120
1	121 122
5	126
8	121
9	122
1.5	123
"120	126
A	*120/! *123/!2
I	102 *121/% 125/% 126/%
J	102 *122/% 124/%

This shows, for example, that the integer *I* appears on lines 102, 121, 125, and 126; it is unmodified on line 102.

Note that the constant 0 appears in line 120 of figure 1, the constant 1 appears in two lines, and the value of *A* is altered in lines 120 and 123. *A* appears twice in line 123, hence the *2*. (The slashes in the figure are meant to act as separators between the line numbers and the variable type designators.)

The last of the BDS utilities, SUPER RENUM (R), allows the user totally or partially to renumber a program. By including the proper parameters, the user can move lines or blocks of code from one position to another within a program. After they are moved, the lines are renumbered to fall in proper sequence according to the new location.

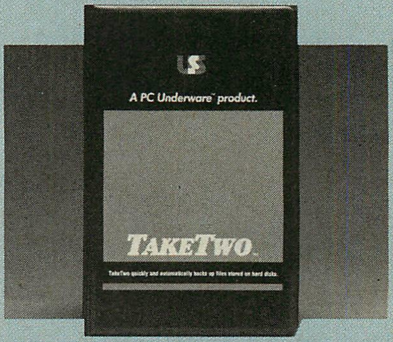
The distribution disk for the IBM version of BDS contains the files needed for each release of DOS/BASIC. The IBM-compatible version of the distribution disk includes files for several different machines and versions of DOS/BASIC. The appropriate file is copied to the BASIC disk, and its name is changed to BDS.COM. The file, along with BASIC, is loaded into memory using the command BDS M BASICA (for BASICA). BDS is then resident in memory and available as long as the user does not exit to DOS. BDS also can be used in a disk resident mode (BDS D BASICA) if memory is in short supply.

Serious BASIC programmers will find a helpful set of tools in BDS to make their use of the language much more efficient. The documentation is good, but it could use a few more examples. The program is machine-specific, and different versions are available for various IBM compatibles.

—PAUL HULTQUIST

TAKETWO VERSION 1.0
United Software Security
8133 Leesburg Pk., Vienna, VA 22180
703/556-0007

PRICE: $89.95



CIRCLE 350 ON READER SERVICE CARD

TakeTwo, a hard-disk backup program, is part of a new genre of accessory software. Programs of this type satisfy a single, well-defined need, usually for less than $200. Other popular examples of accessory software include Sideways from Funk Software, Inc., Borland's SideKick and Turbo

Lightning, and Fastback, another backup utility, from Fifth Generation Systems. (Fastback was *PC Tech Journal*'s October 1985 Product of the Month.)

TakeTwo can be compared to the five disk backup programs reviewed in the March 1986 issue of *PC Tech Journal* ("Backup Utility Performance" (Steven Armbrust and Ted Forgeron, p. 78): BackTrack from Tallgrass Software Technologies; BAKUP from Software Integration, Inc.; DSBackup from Design Software, Inc.; Dump/Restore from Cogitate, Inc.; and Fastback. Like these programs, TakeTwo is a replacement for DOS BACKUP and RESTORE and is much faster and more convenient to use than those commands.

TakeTwo features a unique system configuration screen. In addition to setting up the routine configuration parameters, the user can choose from a variety of noises that will signal when it is time to insert a new backup diskette. The choices include the common PC speaker beep, a trumpet charge, doorbell chimes, and a siren. In addition, with the system configuration screen, a user can instruct TakeTwo to perform an incremental backup daily and a full backup once a month. When TakeTwo is invoked, it checks the date and decides which type of backup to perform.

While backing up files, TakeTwo makes good use of the screen display. In addition to letting the user know which file and directory currently are being copied to disk, TakeTwo displays a kilobyte-per-minute indicator as well as the percentages of disks, directories, files, and bytes that have been copied. The program also displays bars that slowly change to high intensity as the diskettes are filled.

TakeTwo allows the user to select options from menus or to run the program from a command line. It also can be run from a batch file.

TakeTwo does not provide on-line help or printed reports, but it does provide an alternative to a print-out: the history screen, which displays a complete file catalogue with backup dates. TakeTwo uses the standard DOS COPY command; as a result, file compression is not possible. TakeTwo can perform read-after-write verify on both BACKUP and RESTORE by typing the DOS command VERIFY ON. TakeTwo disks can be reused without reformatting.

Although the user's manual does not mention that TakeTwo is copy protected, the fact that it is became apparent during installation—the program uses the Softguard SuperLok technology. TakeTwo does not require a key disk inserted in drive A: to start the program, but the user is limited to two installs, and, worse, the user's manual does not describe how to do an uninstall if the user needs to move the program to another hard-disk drive. The uninstall procedure is possible by running the INSTALLH.COM program, which is included on the TakeTwo distribution disk, with a /U parameter (for uninstall); this rids the hard disk's root directory of SuperLok's hidden files and increases TakeTwo's remaining install count from one to two.

TakeTwo uses standard DOS formatted diskettes. If the user runs out of formatted diskettes in midstream, TakeTwo will format the disks as they are inserted without forcing the user to start the backup again. At the beginning of a backup session, TakeTwo estimates how many diskettes will be needed. However, this count is not always accurate. During testing, TakeTwo claimed that fourteen 1.2MB diskettes would be needed to back up a 20MB, 6-MHz AT; the procedure actually required 16.

The program has three modes of operation: full, modified, and automatic. In full mode, all files on the hard disk are backed up regardless of whether or not they have been changed since the last time they were backed up. Modified mode backs up files that have changed since the last backup. In automatic mode, TakeTwo figures out whether it should do a full or a modified backup and prompts the user to insert the appropriately numbered disk.

Like BAKUP and Fastback, TakeTwo stores all backups (full, modified, or automatic) on a single set of numbered disks. This relieves the user from the task of keeping track of multiple sets of disks and worrying about which set has the latest copy of a file.

Speed is of great importance for disk backup programs. The faster the program, the more often the user is likely to use it. The benchmark used in "Backup Utility Performance" was used again to measure performance for this review. The benchmark tests how long it takes to back up a 20MB, 6-MHz AT with 1,895 files and 62 directories. The test was performed on a machine running DOS 3.1 with BUFFERS=20.

TakeTwo does not set any performance records; the program took 28 minutes 46 seconds to execute the benchmark. This time beats that of BackTrack, BAKUP, and Dump/Restore, but it is almost three times slower than that of

Fastback. An incremental backup of a single 38KB file took 3 minutes 8 seconds. Of that time, TakeTwo spent more than 2½ minutes scanning the directory structure on the disk to see if any files needed to be backed up. Another 20 seconds was spent updating the catalog on both the backup disk and the hard disk. During the catalog update, TakeTwo warns the user not to remove the backup diskette.

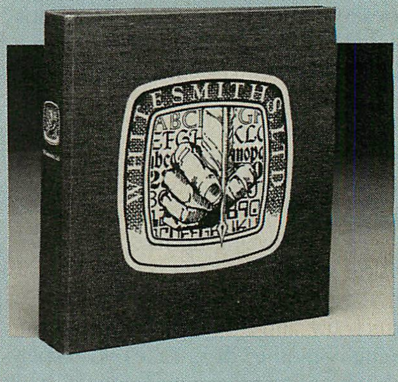
TakeTwo is a solid product. It is simple to use and rich in features. It compares favorably with Fastback because, while Fastback is three times as fast, it is also three times as expensive.

—TED FORGERON

WHITESMITH'S C COMPILER 3.01

Whitesmith's Limited
97 Lowell Road, Concord, MA 01742
617/369-8499

PRICE: $1,000



CIRCLE 352 ON READER SERVICE CARD

While most vendors are working to establish a C standard, the Whitesmith C compiler maintains its own libraries and operating system interfaces. Some programmers appreciate the low-level power of the Whitesmith implementation, but many see only the compatibility snags.

When the Whitesmith compiler is compared with other C compilers currently available on the market (see "The State of C," William J. Hunt, January 1986, p. 82), several fundamental differences are evident. (See table 1 for the basic specifications of the Whitesmith compiler and table 2 for a summary of the compiler's functions.)

More so than other compilers, Whitesmith's C assumes that the user is a professional software developer who understands how the compiler is constructed. Furthermore, the Whitesmith implementation places greater emphasis on portability than do the others. The

TABLE 1: *Whitesmith's C Specifications*

VERSION TESTED	3.01
SUPPORTED ON OTHER SYSTEMS	See text
CROSS-COMPILER HOSTS	See text
AVAILABILITY OF ADD-ON LIBRARIES	Poor
MINIMUM DISK SPACE REQUIRED	1.8MB
MINIMUM RAM	256KB
SUPPORTS FULL LANGUAGE	Yes
FULL STANDARD LIBRARY	Yes
PC-SPECIFIC FUNCTIONS	Yes
ASSEMBLY LANGUAGE INTERFACE	Yes
COMPATIBILITY	
MASM	No
LINK	No
SOURCE CODE	
Start-up sequence	No
Library functions	No
MEMORY MODELS	
Large	Yes
Medium	Yes
Compact	Yes
Small	Yes
.COM	Yes
OTHER PROGRAMS INCLUDED	
Librarian	Yes
Assembler	Yes
Linker	Yes
Source-level debugger	No
MAKE	Librarian
Other	PR, etc.

These specifications can be compared with those for other C compilers listed in table 1 in "The State of C" by William J. Hunt (January 1986, p. 84).

Whitesmith's C is portable to Whitesmith compilers on 30 different machines. The three complete libraries included with the compiler account for its 1.8MB size.

compiler is available in more than 30 different versions, each designed for various combinations of host and target computers. Host operating systems supported include UNIX, VAX VMS, CP/M-80 and CP/M 68K, AMOS, and Idris, as well as DOS.

In addition, differences exist between Whitesmith's C and other C compilers in the runtime environment and the construction of the compiler and development tools.

The defined runtime environment for Whitesmith's C is a generic set of functions that map across all the target computers supported. This creates confusion because the names of many of these functions do not correspond to the names of the UNIX equivalents. In addition, some basic functions such as OPEN, CLOSE, READ, and WRITE have the same names as the UNIX equivalents, but different parameters. As a result, a UNIX-based C program is not easily converted to the Whitesmith environment, especially if the program performs nonbuffered I/O using READ(), WRITE(), or other low-level services.

Whitesmith uses a prototype file with the compiler's development tools to tell the preprocessor, the compiler stages, and the linker which files to process and which options to use. Most other C compilers use switches to accomplish this task. The prototype file is more flexible than switches, but also much more difficult to use. The syntax of the file resembles that of a job control language used on a mainframe that is a myriad of possible options, flags, and steps. In a team development environment, several prototype files might be maintained to support different compilation options. With a single user or small group, this maintenance task is much more trouble than it is worth.

Whitesmith's C is contained on seven disks that are shipped in two fabric-covered looseleaf binders. The first six disks contain the current release of the compiler; the seventh contains replacement files that can be used to correct program bugs. The documentation in the first binder covers the basics of the compiler's operation and support for C. The second binder documents the PC-

TABLE 2: *Compiler Features*

COMPILER OPERATION	
Single-step compile command	Yes
Compile and link	Yes
Accepts list of files	Yes
Accepts wildcards	No
Lists preprocessor output	Yes
Lists assembler output	Yes
Line numbers in error messages	Yes
Header file search list	No
Flexible disk file layout	No
C LANGUAGE EXTENSIONS	
Embedded assembly language	No
Void function returns	Yes
Enumerated types	Yes
Structure assignment, etc.	Yes
Function argument checking	Yes
LIBRARY EXTENSIONS	
Math functions (sqrt, exp, etc.)	Yes
Unbuffered file I/O	Yes
Keyboard input (low-level)	Yes
PC screen output (cursor control, attributes, scroll)	No
Execute programs/DOS (exec/fork and system)	Yes
DOS services (date, time, etc.)	Some
PC-specific functions	Some
UNIX-compatible functions	Some
Error recovery (setjmp(), longjmp())	Yes
FILE I/O	
Redirection	Yes
Full path names	Yes
DOS 1.1 support	No
DOS 3.1 file sharing	No
Record locking	No
ASCII/binary mode	Yes
MEMORY USAGE	
Overlays	No
Default stack size	Yes
Stack size settable	Yes
Stack overflow checking	No
8086 FAMILY SUPPORT	
Byte/word alignment	Yes
80186/80286 support	No
8087/80287 support	Yes
Automatic sensing	Yes
ROM support	Yes

These compiler features can be compared with those for other C compilers listed in table 2 in "The State of C" by William J. Hunt (January 1986, p. 86).

The Whitesmith's C compiler includes three complete libraries. They are full featured but not compatible with standard C libraries.

specific parts of the compiler and the library. Because of this division, users may have difficulty deciding in which binder to look for the answer to a specific question. Fortunately, the installation instructions are provided separately. Overall, the documentation is complete (see table 3).

The development programs, the header files, and the libraries are each kept in separate directories. The total space required by the program is 1.8MB, about twice the average of the compilers reviewed in "The State of C." Whitesmith's C requires so much memory because it contains three complete libraries: ANSII library, extended ANSII, and a portable library.

The system consists of a three-pass compiler, an assembler, a linker, and an object format translator. All six of these components are necessary to producing an .EXE file. A lister and various object library utilities also are included. The assembler and linker use proprietary object files that are not compatible with the LINK.OBJ format. Further, the assembler source format is not compati-

TABLE 3: *Documentation Quality*

INSTALLATION	
Packing list	Fair
File inventory	Fair
Key files described	Poor
Quick step-by-step procedure	Good
Instructions for floppy and hard-disk configurations	Good
List changes from last version	Good
SET-UP	
Set-up assumptions described	Good
Notes on RAM/second hard disk	Fair
OPERATIONS EXPLAINED	
Compile options	Poor
Compiler error messages	Fair
Linking C programs	Poor
Runtime error messages	Fair
Runtime options	Fair
LANGUAGE/LIBRARY SPECIFICATIONS	
Deviations from Kernighan and Ritchie standard	Fair
Data type representation	Fair
Memory models and memory layout	Good
DOS and PC-specific features	Poor
ASSEMBLY LANGUAGE INTERFACE	
Segment, group, and class specification	Fair
Standard prologue, epilogue	Fair
Instruction formats for args, public, extern, struct	Fair
Return value conventions	Fair
Complete examples	Poor
FILE I/O	
Redirection	Fair
Console I/O	Fair
Device I/O	Fair
Buffered versus unbuffered	Fair
ASCII versus binary modes	Fair
LIBRARY DOCUMENTATION	
Average lines per function	30
Cross-reference information	Good
Functions in table of contents	Good
Examples of use	Poor
MANUAL ORGANIZATION	
Detailed table of contents	Good
Index with functional entries	Good
Order of function documentation	Poor
OVERALL RATING	Fair

These notes on the documentation for Whitesmith's C compiler can be compared with those for other compilers listed in table 3 in "The State of C" by William J. Hunt (January 1986, p. 88).

The documentation included with Whitesmith's C compiler is complete and technical. A discussion of ANSI extensions and portability guidelines is provided.

ble with MASM. These incompatibilities severely limit add-on library support.

The benchmark programs for this evaluation were run on an Alpha Micro Workstation with 512KB and a 10MB hard disk (see table 4 for benchmark results). CONFIG.SYS contained FILES=20 and BUFFERS=20. Because so many phases are required for compilation and linking, and because a prototype file and a driver control the entire process, compilation and link times for the Whitesmith compiler are slow. In particular, the link times are almost twice as long as those for other C compilers run on the PC.

However, once compilation is completed, the actual code produced by the compiler falls well within the range of acceptable speed and size standards for C compilers on the PC. The exceptions are the character I/O functions **getc()** and **putc()**, for which the execution times are slower than average. Register variables are supported and, when used, increase execution speed.

Whitesmith's C offers two definite advantages. First, it supports, in addition

TABLE 4: *Performance Benchmarks*

COMPILE TIMES	
60-line file	47.6
150-line file	98.0
500-line file	159.5
LINK TIMES	
1 object file	69.3
6 object files	73.3
PROGRAM SIZES (bytes)	
Sieve	17,104
Pentathlon	19,536
GENERAL OPERATIONS	
(small/large model)	
Function calls (Fibonacci)	29.5/30.8
Integer arithmetic	39.2/41.6
Long arithmetic	115.2/120.1
Subscripts (character count)	24.1/33.9
Pointer use (string copy)	44.0/58.7
With register variables	31.3/58.8
Eratosthenes sieve	25.3/29.3
With register variables	18.7/20.3
FILE I/O	
(small/large model)	
Read/write	
Floppy to floppy	7.5/8.0
Hard disk to hard disk	3.2/4.6
Getc/putc	
Floppy to floppy	116.4/119.6
Hard disk to hard disk	46.7/52.3
Floating-point operations	
(small/large model)	
Add/multiply (dot product)	152.8/161.9
Exp/log	150.8/150.2
Sin/tan (trig functions)	215.2/205.0

Because Whitesmith's C is too large for effective floppy-disk compilations, only hard-disk compilation times are listed.

Benchmarks were run on a PC/XT compatible with a Norton speed index of 1.0. Care should be taken when comparing these benchmark times to those listed in table 4 in "The State of C" (William J. Hunt, January 1986, p. 90), which were reported for an IBM PC/XT.

The benchmark tests, designed for standard C compilers, were modified considerably to compile under Whitesmith's C. The benchmark times are quite respectable.

to the five standard memory models (small, large program, large data, large, and .COM), an extended small memory model. This sixth option allows not only the small data items to be linked into groups, but a few large data items to be linked as well. Programs that have only a few large data structures can be compiled without every data item being referenced through a far pointer and, as a result, without sacrificing any speed. This flexibility is not available in other C compilers.

In addition, the extended small memory model provides a **#pragma** statement that allows the user to control the allocation of data and functions within the program. For example, **#pragma** can make a specific function *far callable* or *interrupt callable*. When the program is completely linked, the appropriate calling sequences are generated whenever the function is referenced. This is useful for code that interfaces with the PC's operating system or hardware at an extremely low level.

As a second advantage, Whitesmith's C offers support and documentation of the ANSI extensions to the C language. Thus, assuming the coding rules are followed and the appropriate header files are used, programs written in Whitesmith's C are portable to other hardware systems running the ANSI standard with very little change. The manual details portability considerations for writing ANSI-standard programs.

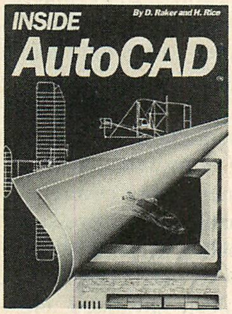
These two advantages are minimized by the product's needless incompatibilities. The user is left wondering why Whitesmith insists on marching to the beat of a different drummer.

—MARTY FRANZ

Practical CAD

This hands-on tutorial for Autodesk's AutoCAD is a must for newcomers and worthwhile for veterans.

Inside AutoCAD
Daniel Raker and Harbert Rice (New Riders Publishing, Thousand Oaks, CA 1985) 308 pages, paper, $27.95



Regular reviewers of CAD/CAM/CAE software often are asked where users might find a how-to book on Autodesk's AutoCAD (or Personal CAD Systems' p-CAD or T & W Systems' VersaCAD or other such packages). Dealer demonstrations and demonstration disks are not able to provide users with an on-going reference for how to do their job with the specific software they have chosen.

Book publishers and bookstores like to concentrate on computer books about programs that sell hundreds of thousands of copies, not those that sell only tens of thousands. Books on CAD are rare, and even if they were not so rare, they would be lost in the crowd of books on spreadsheets, word processors, and programming languages. (If they were to judge from the material on some bookshelves, readers might think that publishers believe there are more C programmers than architects, engineers, and plumbers combined.)

Inside AutoCAD is the book many students of CAD have been looking for. Instead of a treatise on the mathematics of computer graphics, this book is about a real program, AutoCAD (version 2.0). Written in a tutorial style, it remains concentrated on its subject throughout. Readers will find no sections on how to operate the computer or on the history of computers or on the joy of computing. The authors wisely assume that anyone who buys the book already knows how to turn on the computer and so on, and that the real interest is in using AutoCAD in gainful employment—not for the thrill of drawing circle and lines on a screen.

Both authors are well qualified on the subject of CAD, particularly in the practical uses of CAD. Daniel Raker is a consultant well-known to design professionals through his *A/E SYSTEMS REPORT*, a monthly newsletter that addresses the issues of design automation, and through his column "CAD Angles" in *Plan and Print* magazine. Harbert Rice owns the company that published the book and is a veteran in the fields of pattern recognition and simulation.

Inside AutoCAD is well-organized. The introduction explains the book's layout, discusses its intended use, and defines several key assumptions; it includes a summary of each chapter and an explanation of the intent of each. Those who generally cannot force themselves to read introductions should make an exception in this case.

As most AutoCAD users know, the AutoCAD manual is not organized as a tuturial, but as a reference manual, and even that theory of its organization commonly escapes the newcomer. *Inside AutoCAD* is organized as a teaching guide, and the sequence of material is ideal for either self-instruction or the classroom. The authors know that the new user will wish immediately to draw a few circles and lines—play around, that is—so they begin by leading him into doing just that. However, only a few pages are devoted to play, then the book turns to setting up AutoCAD for productive work.

The first chapter covers the menu system and the concepts of layers, scales, limits, and units, all of which are key concepts for production work. In the second chapter, the display controls (ZOOM, PAN, and VIEW) and drawing aids (SNAP, ORTHO, GRIDs, object snap) are examined. Again, these concepts are essential to effective use of the program, even though it is possible to make drawings without fully understanding them.

Only by the third chapter do Raker and Rice arrive at the point at which most users are tempted to begin, that is, drawing lines and circles. In full, this chapter, called "Graphics Primitives," covers lines, arcs, circles, and text.

Chapter 4 covers editing a drawing and explains the significance of the editing commands. The authors concede that a swift draftsman often can draw lines faster than a CAD operator, but that the AutoCAD user can revise drawings much faster than his conventional counterpart. Readers may not necessarily agree with this statement, but the point is that AutoCAD provides a host of tools not available to the conventional draftsman: COPY, MIRROR, FILLET, CHAMFER, ARRAY, MOVE, CHANGE, REPEAT...ENDREP, BREAK, and ERASE. The authors insinuate that the editing commands allow the AutoCAD draftsman to create a number of drawing objects with a minimum of original entry, but they concentrate on the idea that the real advantage of AutoCAD is in posting changes. Some CAD users believe that they must learn to think in new terms when designing with AutoCAD or any CAD system, and the editing commands are the key to this new mode of thought. Regardless, *Inside AutoCAD* provides excellent coverage of the editing commands.

One of AutoCAD's strongest features, BLOCKS, is given attention in the fifth chapter. A symbol library is a sine qua non for a CAD program. This chapter discusses the process of defining and inserting blocks and includes an extensive exercise in the use of blocks to create a drawing that would be very time-consuming to develop without the use of blocks. Attributes are covered in a later chapter.

Chapter 6 introduces the commands used to dress up drawings, including primitives not covered in the

earlier chapters, such as TRACE and SOLID, and the HATCH and SKETCH commands. These can relieve the stark look of CAD drawings and convey essential information in some fields.

The treatment of plotting in chapter 7 covers all but the most demanding situations and includes a checklist for plotting that would be useful in conjunction with any CAD program. A full 25 pages, the chapter begins with an exercise that the authors began in the chapter on blocks.

In chapters 8 and 9, Raker and Rice cover automatic dimensioning and the use of attributes. Automatic dimensioning is an essential capability for any CAD program, and the ability to attach attributes to drawing symbols is the key to the second D in CADD. Both subjects are covered thoroughly.

The final chapter covers the subject of customizing AutoCAD, through the use of menus and script files. Several examples are presented that should enable any serious user to write his or her own custom menus. Not covered are the use of variables and expressions or the use of AutoLISP, which is introduced in AutoCAD version 2.18.

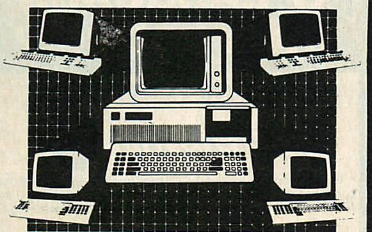
The book is nicely illustrated, and all of the illustrations obviously were prepared with the aid of AutoCAD (indeed, any other approach would have aroused suspicion). Commands are shown with examples that include screen displays, prompts, and menu flow charts. Thus, whether the user prefers to use the screen menu or to type commands in from the keyboard, the command syntax is supplied. The book does not include examples of using digitizer menus, but this is not a serious omission.

Each chapter opens with a diagram of the standard menu structure as it relates to the commands covered in that section and closes with an advanced AutoCAD drawing. Taken collectively, these drawings show the capability of AutoCAD in a variety of fields.

The book's tone is appropriate for a range of AutoCAD users, from the novice to the advanced draftsman. The veteran will find some worthwhile nuggets, while the beginner should consider this book a required companion to the AutoCAD manual. Raker and Rice have put together an excellent presentation of AutoCAD's facilities. (*PC Tech Journal* reviewed AutoCAD in the January 1986 issue, "Drafting by Design," Victor E. Wright, p. 50.)

—VICTOR E. WRIGHT

More Users For Less Money!
Digitrol's software solution to multi-user.



pcSHARE
MULTI-USER O/S

pcSHARE allows your IBM-XT/AT or compatible to support up to 5 users running 1-2-3, dBASE, WordStar, and more on inexpensive serial CRTs. For software developers, pcSHARE efficiently runs compiled Basic, Pascal & C programs with full DOS 3.0 compatible record locking.

DIGITROL COMPUTERS INC.
440 Phillip Street
Waterloo, Ontario, CAN. N2L 5R9
(519) 884-4541

IBM-XT, Lotus 1-2-3, dBASE, and WordStar are trademarks of IBM Corp., Lotus Development Corp., Aston-Tate, and MicroPro International, respectively.

CIRCLE 384 ON READER SERVICE CARD

MYSTIC PASCAL
Fastest Compiler on Earth

NOW ONLY $16!!!

- 640K code, data and stack
- incremental compiler runs in background while you are editing
- effective compiler speed greater than 100,000 lines/min
- immediate execution of Pascal statements
- true multi-tasking for up to 100 Pascal procedures
- full screen editor
- complete Help windows

disk with condensed manual to print out $16—printed manual $16—both $32—for IBM PC 256K Visa/MC/COD—add $2 shipping, COD $5, overseas $10

Mystic Canyon Software
P.O.B. 1010
Pecos, NM 87552 505/757-6344

CIRCLE 388 ON READER SERVICE CARD

DISKETTES
WHY PAY MORE??

39¢ 5¼" DS/DD
SOFT SECTOR
BULK QUANTITY 100

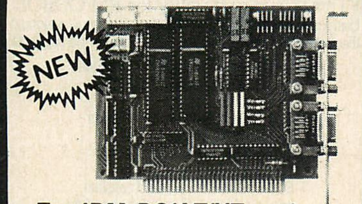
Each diskette 100% certified at manufacturer.
LIFETIME WARRANTY

ACT NOW!!
These are not seconds.
Money back guarantee!
No questions asked.
Offer Expires 6/30/86

TO ORDER CALL:
1-800-634-3478
In Ohio: (614) 481-4417

MEI
1555 West Lane Ave.
Columbus, OH 43221

RS-422
Communications Board



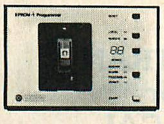
- For IBM-PC/AT/XT and compatibles
- Dual RS-422 serial interface
- Programmable to 56k baud
- Differential drivers to 4000 ft.

$345.00

QUA TECH, INC.
478 E. Exchange St. Akron OH 44304
(216) 434-3154 TLX: 5101012726

CIRCLE 386 ON READER SERVICE CARD

NEW EPROM PROGRAMMER



EPROM 1
$495

*Programs all 28-Pin, 5-Volt EPROMs and EEPROMs.
*Perfect for engineering workstations, field service and small production runs.
*Simple keyboard and display for stand alone duplication.
*Interface to your computer via an RS232C port.
*PC control software included at no extra charge.

TO ORDER PHONE:
800-325-6028 Outside California
916-885-7262 Inside California
VISA AND MASTERCARD ACCEPTED

INTERNATIONAL MICROSYSTEMS INCORPORATED
11554 C Avenue, Auburn, CA 95603

CIRCLE 389 ON READER SERVICE CARD

FREE DISK SPACE

THE FILE STORAGE METHOD USED BY PCTECHLINE.

The ARC file archive utility can save up to 50% on disk storage, and modern transfer time.

- **Not copy protected**
- **Program Sources Included**

Only $50

"A sophisticated and imminently useful product."
—PC Week

SYSTEM ENHANCEMENT ASSOCIATES
21 New Street, Wayne, NJ 07470
(201) 473-5153

CIRCLE 394 ON READER SERVICE CARD

TECH JOURNAL

TECH MART
TECH BOOK

THE CLASSIFIED ADVERTISER'S FORMULA FOR SUCCESS IN SELLING TO TECHNICAL EXPERTS IN THE IBM MARKETPLACE.

FOR RATES & INFO CALL: (212) 503-5115

XT new & AT HD TEST

HDTEST formats and tests hard drives in PC, XT, AT, and true compatibles. Written for production quality control; you know it's fast and thorough. Menu-driven operation and context sensitive help windows make life easy when installing or reformatting hard drives. Flag known bad tracks and do a comprehensive surface scan to find unlisted defects. You can modify controller and test setup, load/save setup and defect files. Works with WD, Xebec, OMTI, DTC, and Adaptec controllers. Free PARK included! Call for more information. $99.00
GALLERY 6 DISTRIBUTED BY PROTO PC, INC.
2439 FRANKLIN AVENUE ST. PAUL, MN 55114
612-644-4660 TLX910-380-7623

CIRCLE 392 ON READER SERVICE CARD

Get the whole story on graphics terminal emulation.

To find out more about software that lets your PC emulate TEKTRONIX™ 4105/6/7/9 and DEC VT100™ terminals, call or write:

GRAFPOINT
4340 Stevens Creeks Blvd., Suite 280,
San Jose, CA 95129 (408) 249-7951

CIRCLE 387 ON READER SERVICE CARD

ASMLIB

ASSEMBLY PROGRAMMING LIBRARY

- NO ROYALTIES REQUIRED
- Graphics - Color, Herc, Monochrome, EGA
- Floating Point Trigonometry and Arithmetics
- 8087 Supported but not required
- ASCII to Numeric conversions
- Formatted Output (PRINT USING)
- Console I/O with Windowing
- DOS Shells
- Sound Generation
- Plus much, much more

ASMLIB has over 170 functions for the IBM PC/XT/AT or compatible under DOS 2.00 or newer. Supplied in MS Assembly language source code on 3 DOS Diskettes with 215+ page reference manual.

All for only ... $149.00 ppd

BC ASSOCIATES
13073 Springdale Street, Suite 134
Westminster, California 92683
(714) 741-3015

CIRCLE 376 ON READER SERVICE CARD

64K ■ 128K ■ 256K DRAMS

80287-8 ■ 80287-3
8087-3 ■ 8087-2
8087-1

BITTNER ELECTRONICS

899 SOUTH COAST HIGHWAY
LAGUNA BEACH, CA 92651
(714) 497-6200

CALL NOW FOR **FREE** CATALOG

CIRCLE 377 ON READER SERVICE CARD

Learn the C Language Interactively

Step-by-Step With *Introducing* **C**

Introducing C is a powerful C language training system that combines a thorough, self-paced manual with a unique C INTERPRETER to provide a fast efficient method of learning C.

A COMPREHENSIVE APPROACH

Introducing C covers all the essential elements of C. The Interpreter utilizes standard K&R syntax and operators – full structures and unions, arrays, pointer and data types. It provides the standard I/O library and an extended graphics library. The package has many user oriented features including a screen text editor and extensive error diagnostics (includes dynamic trace capabilities).

INTRODUCE YOURSELF, TODAY

Introducing C is available for immediate delivery. System requirements are IBM PC, XT or AT with one disk drive and 192K bytes of memory. Priced $125.00. Not copy protected.

TO ORDER CALL 800-922-0169

COMPUTER INNOVATIONS, INC.
980 Shrewsbury Ave., Tinton Falls, NJ 07724 • (201) 542-5920

CIRCLE 378 ON READER SERVICE CARD

SUPER LOCK™

IBM PC, XT and AT programmers:

Don't give away results of your effort! Don't waste your money buying costly fingerprinted diskettes!

You can have your programs automatically protected against piracy!

- **No special training needed!**
- **No confusing instructions!**
- **No discouraging procedures!**

No Sales Tax on orders outside N.Y. State

$ 180

YETIWARE CORP.
P.O. BOX 21152
MID TOWN STATION
NEW YORK N.Y. 10129

DEALER INQUIRIES INVITED

CIRCLE 380 ON READER SERVICE CARD

FORTRAN or PASCAL PROGRAMMER?

READ THIS AD

Sort RM/COBOL Files

QISORTC

Finally, a sort utility for RM/COBOL. Using efficient algorithms in assembly language making it the fastest sort on the MS-DOS market. Qisortc supports:

- RM/COBOL Version 1.5x & 2.xx files
  - Single / Dual File Indexed
  - Relative and Sequential
- ASCII, COMP-1, & COMP-3 fields
- 32 sort control fields
  - ascending / descending
- 32 record selection fields
  - EQ:NE:GT:GE:LT:LE field:constant
  - implied AND, OR may be specified
- 32 output fields or total record
- CALL from RM/COBOL, when provided subroutine is linked with runtime
- Use pathnames and return codes
- Keyboard or Parameter file
- Requires MS-DOS ver 2.xx or above
- Only $95.00 - COD, VISA, or M/C.

Quantumn Information Systems, Inc.
145 N.W. 85th St., Suite 103
Seattle, Washington 98117
(206) 789-2888 Telex 4941041

CIRCLE 381 ON READER SERVICE CARD

PERFECT COPIES GUARANTEED

Why risk duplicating your important programs on your computer, when *our* equipment is designed solely to duplicate disks and verify their perfection 100%?

Over 600 formats. 3½", 5¼" and 8". Plus serialization, copy protection, labeling, packaging, shrink-wrapping and *fast*, personalized service.

Western Transdata Inc.
DISK DUPLICATION

1701 E. Edinger Ave.
Building A4
Santa Ana, California 92705
(714) 547-3383 (Collect)

CIRCLE 382 ON READER SERVICE CARD

GRAPHICS

- Text/graphics generics
- 2D interactive
- 2D plots (full support) including auto-scaling, auto-axis generation, auto-labeling, tabular/log/parametric curves, splines, bars, pies, you name it, we have it.
- 3D Plots incl. 2 hidden line removal options—not just old-fashioned wire frame.

PEN PLOTS

- Standard plotter primitives plus
- FULL 2D support plus
- Interface to screen graphics,
- Limited 3D.

Clear and complete documentation.

GRAFMATIC $135.
PLOTMATIC 135.

MICROCOMPATIBLES
301 Prelude Drive
Silver Spring, MD 20901
(301) 593-0683

CIRCLE 383 ON READER SERVICE CARD

REPRINTS AVAILABLE

Quantity reprints of articles appearing in **PC Tech Journal** are available and will be prepared to meet any special requirements. Inquiries should be directed to Eileen Pfeiffer, Reprints Dept., Ziff-Davis Publishing Co., 1 Park Ave., New York, New York 10016, Phone 212-503-5447.

PC TECH JOURNAL

TECH BOOK

Accessories/Supplies

•• SOFTWARE PUBLISHING ••
GDS offers a wide variety of services that will help get your software to the market. Address your needs with GDS.
- IBM style cloth/vinyl 3-ring binders/slips
- Labels, sleeves, disk pages ...
- Disk duplication with 100% verification
- Bulk diskettes
- Shrink wrapping and assembly
- Quick turnaround

A well-packaged product can make the difference in making a sale. Call us NOW. VISA/MC
Glenco Development Systems
3920 North Ridge Avenue
Arlington Heights, IL 60004
(312) 392-2492

Business Opportunities

INCORPORATE!
*Save Taxes!
*Protect Your assets!
*Free Booklet!
*Free Forms!
*Save $— Do It Yourself!
*No Legal Fees!
WRITE OR CALL TOLL-FREE TODAY!
HARVARD BUSINESS SERVICES, INC.
3151 Summerset Rd.
Wilmington, DE 19810
800-345-CORP

Personal Computer Owners
CAN EARN $1,000 TO $5,000
monthly selling simple services performed by their computer. Work at home-in spare time. Get free list of 100 best services to offer. Write:
C.I.L.B.Q.
PO Box 60369
San Diego, CA 92106-8369

Hardware/Add-On Boards

SPEECH SYNTHESIS
SynPhonix: TRUE Unlimited Speech Synthesizer for IBM-PC/XT/AT/jr & compatibles. This low power short card includes an SSi263 speech chip, amplifier and speaker. Software includes Text-to-Speech, Phonetic Editor, Talking Clock & demos. Can be programmed with BASIC and other languages. Prices start below $200.

SynPhonix™
Electronic Speech Articulator

Artic Technologies
2234 Star Court
Auburn Heights, Michigan 48057
(313) 852-8344

RS-232/RS-422 Communications
Versatile communications and I/O boards.
—Four port RS-232 with DOS drivers $349
—RS-422/485-serial interface $149
—RS-232/422 with 24 parallel I/O $199
—RS-422 synchronous with DMA $299
—Delivery from stock
—Custom designs available

adams

Adams Incorporated
P.O. Box 17525
Greenville, SC 29606
(803) 297-9630

FIXED DISK BIOS/BOOT
FiXT boots from most popular Hard Disks—DAVONG, TECMAR, IOMEGA, GT LAKES, etc. Adds XT-like BIOS interface to your disk for PC. Security, multiple volumes, removable media support optional. No-slot plug-in installation. Specify controller and computer with order. $80-$95. Add $3 shpg., CA tax.

GOLDEN BOW SYSTEMS
2870 Fifth Avenue
Suite 201
San Diego, CA 92103
(619) 298-9349

NEW!!! FOR THE OLD 64K PC
PC MultiPak MAX 576K Multifunction Card. Expand your old 64K IBM to 640K. Includes 576K memory, parallel port, serial port, clock, ramdisk & ramspool software, 2 year warranty. $449 with 576K installed. Trade in discounts available for your current memory card Mfg. in USA. For more information call or write
Indigo Data Systems, Inc.
P.O. Box 57888
Webster, TX 77598
(713) 488-8186

ACQUIRE DATA WITH YOUR PC
*16 to 64 differential A-D, up to 8 D-A
*24 to 48 digital I/O, 24 event counters
*Processor and 8 to 40K RAM on board
*Floating-pt processing, Battery backup
*Linerization, 4 selectable gain ranges
*Menu-driven software. Programmable.
*Systems start at $895, 1 year guarantee
WESTERN TELECOMPUTING
202 E. Kagy Bozeman, MT 59715
(406) 586-1511, or touch-tone, toll-free
1-800-423-0587, then 328 222.

Hardware/Disk Drives

8 INCH DISKETTE SYSTEM
Read, write and format diskettes from IBM mainframes, minicomputers, data entry equipment, etc. Complete easy to use software handles EBCDIC conversion. Can read and write CP/M 8-inch diskettes (many formats). You also can use 8-inch drives for PCDOS files; 1200 KB per diskette! $1150 complete.
MicroTech Exports
223 Forest Avenue
Palo Alto, CA 94301
(415) 324-9114

Hardware/Peripherals

SCSI-PC HOST ADAPTER
Plug & play installation for SCSI Optical/Hard disk and tape peripherals. Wide selection of SCSI peripherals selectable by manufacturer and model No. via menu window. Absolutely NO driver software need be written by the user. Price is $149. for board, SCSI bios prom and floppy based configuration data.
Advanced Storage Concepts, Inc.
9660 Hillcroft #325
Houston, TX 77096
1-800-423-9175 (713) 729-6388

640K YOUR MOTHERBOARD!!
Put 640K of RAM on Your Motherboard! NO SLOT! NO ADD-ON BOARDS!—NO SOLDERING! Complete w/ALL Hardware, Memory Chips and Instructions. Easy PLUG-IN Installation. Avail for: IBM XT, IBM Portable, COMPAQ & COMPAQ PLUS Portables. Only $144.95 + $5 S/H, Specify computer when ordering. Dlr. inquiries invited. CK/MO/VISA/MC
J S & J Software
1281 S. King St. Suite 6A
Honolulu, Hawaii 96814
24/hr Orders: (800) 821-5226 ext. 435

TURBO START YOUR PC!!!
FOR IBM PC's ONLY!! The PC-INMATE, a Firmware ROM Module for the IBM-PC!!! Features: BOOT within 10 SECONDS!!! BREAK the 640K BARRIER!! Access up to 704K of RAM!!! (Lotus, DOS, etc.) SCREEN RELIEF-protects your monitor, and MORE!! Easily PLUGS IN-NO SLOT! Only $49.95 + $3 S/H. Dlr. inquiries accepted CK/MO/VISA/MC.
JS&J Software
1281 S. King St. Suite 6A
Honolulu, HI 96814
24 hr Orders Only: (800) 821-5226 ext. 435

IBM PC/XT/AT ...... SPEEDUP
*OVERTRUSTER: Boots the performance of PC and XT by 280%! 100% software compatible, is switchable between 4.77 and 7.38Mhz. $195.
*SPEEDSWITCHER: This is an economical 6 speed switch (4-11Mhz) for AT's. $89
*XCELEX: Automatic 5 speed switch (6-12Mhz) This unit will switch on the fly and works with ALL IBM AT's including the new Models with speed limiting Rom Bios. $149.
DATA MANAGEMENT SYSTEMS, INC.
1525 Hickory St.
Salina, KS 67401
ORDER HOTLINE (913) 823-6440

CP/M & 1.2Mb AT ON PC
With MULTI-DISK and UniForm-PC use 3.5, 5.25, and 8-inch single and double density CP/M format as DOS diskettes on your IBM PC or XT. Many MS-DOS formats also supported including IBM AT 1.2 Mb, HP-150 and Data General 1. Over 200 formats. Both MULTI-DISK and UniForm-PC for $225. Disk drives & adapter cable also available.
P S Engineering
P.O. Box 51068
San Jose, CA 95151-5068
1-800-369-2398; 1-800-423-7171 in CA.

BAR CODE READERS
- IBM PC/XT, AT, AT&T 6300 & 6300 PLUS keyboard models or RS-232 interface
- NO programming. Reads dot matrix
- Auto-recognition and single code decoding
- Reads Code 39, UPC A/E, Codabar & I 2 of 5
- $635 list, units in stock, 2 year warranty
- Bar code printing software, call for info.

PERCON, Inc.
2190 W. 11th
Eugene, OR 97402
(503) 344-1189

PC SPEEDUP
Up to 45% speed improvement for IBM PC and clones, 100% compatible, easy installation. Kit includes NEC V20 or V30 Chip, Tool, User Guide, Warranty, free Multiuser BBS Access. V20-5 mhz for PC, V20-8 and V30 in stock. V30 replaces 8086. Call for PC-Sprint and VP/M.
From $4 to $89. Tech info 414-242-2165
Exec-PC, Inc.
PO Box 11191 Shorewood, WI 53211 Order on BBS: Orders: 414-242-2173 414-964-5160

Rates and Information

Standard listings consist of a bold lead line (25 characters maximum); 7 lines of body copy (45 characters per line); 4 lines for company name, address and telephone number. $140 per insertion—3 issue minimum. Additional charge for extra lines and company logos. Prepayment and frequency discounts available. American Express, MasterCard, Visa accepted. Copy subject to publishers approval. Send typewritten or printed copy, reproducible logo art (if applicable) and remittance to Kathryn Cumberlander, Classified Sales Manager, Ziff-Davis Publishing Company, One Park Avenue, New York, NY 10016. For additional information, assistance, or to place an order by phone, call collect (212) 503-5115.

TECH BOOK

HARDWARE/SECURITY

STILL WAITING? INSECURE?
QUICKON—Instant Turn-on module—$69.95. LOCKIT I—Password Boot-Up module—$129.95. User selectable PW & Boot-only-from-HD mode. PC RESET—Reboot w/o cycling power—$21.95. Easy to install, No slot, specify PC/XT/AT LOCKIT II—PW protection for subdirectories, for DOS 2.0 & up.—$79.95. MC/VISA
Security Microsystems Consultants
16 Flagg Place Suite 102 TJ
Staten Island, NY 10304
(718) 667-1019

SOFTWARE/BUSINESS

The Andsor Collection
Unique concepts: creates complete, self-contained, window-based data management environment in one DOS file. Simplifies everything. Combine functions to create your own solutions in any application: calculations, database management, modeling, text processing, charts, data analysis, statistics, reports, labels, forms, presentations, mail-merge, etc. Simple enough for a PCjr, sophisticated enough for a PC AT. Superb, 400 page hard-cover manual, with many examples. From simple calculations, files, inquiries, to complex models, data structures, reports: when your favorite data manager/spreadsheet/word processor/integrated system cannot provide the solution you need, remember The Andsor Collection. 60 day money-back guarantee. US$95 + $5 s&h. Visa/MC/AmEx/Chk/MO/COD. Call or write now to order. IBM/PC/XT/AT/PCjr, 128K, one drive or hard disk, monochrome and/or color monitor, DOS 2.0+. Not copy-protected.

The Andsor Collection™

Andsor Research Inc.
181 University Avenue, Suite 1202
Toronto, Ontario, Canada M5H 3M7
(416) 364-8423

SOFTWARE/COMPILER

DeSmet-C - $109
Full K&R C compiler, linker, librarian, fullscreen editor, profiler, overlays, example software, 140 page manual. BOTH 8087 and floating point libraries. OUTSTANDING PRICE/PERFORMANCE. Consistently rated 1st or 2nd in published benchmarks. PC/MSDOS. No royalties on generated code. Newsletter. Updates $20 ea. $109 shipping included. Symbolic (C source level). Debugger Option $50 extra. C Compiler for the Macintosh $150.
C WARE CORPORATION
Dept. PCTJ
P.O. Box C
Sunnyvale, CA 94087
(408) 720-9696

C Compilers for MC 680X0 Now on IBM PC
Use your IBM PC XT to create fast compact code optimized for the Motorola 68000, 68010, and 68020/68881. Complete C development environment including assembler, pre-processor, linking and downline loaders, libraries and utilities. Priced under $1000 on IBM PC. Also available for Motorola/VERSAdos, VAX/UNIX, and VAX/VMS.

Alcyon CORPORATION

Alcyon Corporation
5010 Shoreham Place
San Diego, CA 92122
(619) 587-1155

SOFTWARE/COMMUNICATIONS

COMM. PROTOCOL ANALYZER
Why pay thousands for a protocol analyzer or $250 for a RS232 breakout box? Field proven APA120 software converts a PC. Full function breakout box with LED's switches and jumpers plus line adapter for analyzer. Trigger strings, control line monitoring and more. For serial, async, 9600 baud or less. $495.

BURR-BROWN® BB

Burr-Brown Corporation
P.O. Box 11400
Tucson, AZ 85734
(602) 299-2662

PC SERIAL DATA ANALYZER
Use your IBM PC or compatible to analyze data streams between two serial devices (up to 9600 BAUD). Two windows display each devices trans mition in ASCII or HEX. PC can also act as a terminal for either device. Invaluable tool for debugging serial interfaces. Disk & manual $150.
Triple C Software
2897 SW 13th St.
Fort Lauderdale, FL 33312
(305) 583-0687

Programmable Emulator
Automate your communications. "Hooks" to operate communications functions automatically thru your own programs written in "C," Basic, Pascal. Supports PC to PC to mainframe link up. Memory resident mode, simultaneous COM1 and COM2, X-modem, full featured, VT100 and 13 other terminals. Bluestreak plus $89.95.
Lang-Allan, Inc.
2457 Aloma Ave. Suite B
Winter Park, FL 32792
(800) 541-2255 Tone 677-1539

SOFTWARE/DEVELOPMENT TOOLS

COLOR, CURSOR CONTROL
DATE/TIME, KEYBOARD, PEEK, POKE, COMMAND LINE ACCESS, and more available for IBM/MS FORTRAN, IBM/MS PASCAL, and GENERIC (MS) BASIC on PC/DOS and many MS/DOS machines. No redistribution fees. $47 U.S. MN residents add 6% sales tax. BARTSUBS by BARTSOFT
3210 Lake Shore Boulevard
Wayzata, MN 55391
(612) 473-4032

ROMable CODE on PC!
PCLOCATE allows PC users to develop ROM-based software from MS-DOS "Exe" files. The user specifies the physical location of all segments. Output files are compatible with most PROM programmers. PCLOCATE supports the 8086, 8088, 80186, 80188, and 80286 processors. MC/VISA.
ALDIA SYSTEMS INCORPORATED
P.O. Box 37634
Phoenix, AZ 85069
(602) 866-1786

REALIA COBOL SITES
REALFORM is the Screen Handler you have been looking for. It features interactive screen design, an on-line forms editor, relief of cursor control, expedites screen generation, supports numerous data field types, is easy to use and offers powerful data entry functionality. $395. US.
CHARANDON COMPUTER SYSTEMS INC.
1600 Laperriere Avenue
Ottawa, Ontario, Canada K1Z 8P5
(613) 729-3003

FIRMWARE PRODUCTION ON PC
LINK&LOCATE enables PC users to produce ROM-based firmware for 8086/87/186 from object files generated by C, PL/M compilers & MASM. Provides full control of segments placement anywhere in memory. Supports output of INTEL hex file for PROM programmer, absolute object file for symbolic debugger & ICE, and MS-DOS EXE file. Includes an INTEL compatible linker, locator, librarian and hex formatters. $350.
Systems & Software, Inc.
3303 Harbor Blvd., C11
Costa Mesa, CA 92626
(714) 241-8650

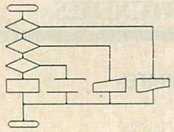
REPORT WRITER
All-language qPLEX-IV lets you select or sort your own special forms, reports, labels, bar graphs or queries. Exports your reports to spreadsheets and wordprocessors. Reads Basic, Assembler, Pascal, C, Fortran, Cobol, ASCII, Data Base Managers and more. Dealer demos available. MS-DOS, XENIX, UNIX
Snow Software Corporation
3330 Fisher Road
Clearwater, FL 33519
(813) 784-8899

C-INDEX
C-INDEX is a state-of-the-art data management library for C. Designed for serious product development. No Royalties. B+Tree, variable length records, multiple keys, and transportable source code. Call for our free info packet.

C-INDEX/PRO	$195 Partial Source
C-INDEX/PLUS	$395 Full Source

TRIO SYSTEMS
2210 Wilshire Blvd. Suite 289
Santa Monica, CA 90403
213-394-0796

FLOWCHARTS BY PROFLO
The fast, friendly, flowchart drawing program for the software professional. PROFLO speeds software development and reduces documentation costs. Lets you create, edit, and print clear, precise flowcharts, quickly and easily. Requires IBM PC (or clone) with DOS, 128K, graphics monitor & graphics printer. Program & Documentation $49.95. Demo Disk $10.00.



Star Systems
28 Topstone Drive
Bethel, CT 06801
(203) 744-0546

SOURCE CODE LIBRARY SYSTEM
TLIB™ stores ALL versions of your source in ONE compact library file, even with hundreds of revisions. Updates ("deltas") faster than many editors load & save files, 5-7 times faster than Unix SCCS. Date & comments for each version, easy retrieval. LAN-shared libraries with PC Network. DOS 2.x/3.x $99.95 $3 s/h VISA/MC.
Burton Systems Software
P.O. Box 4156
Cary, NC 27511-4156
(919) 469-3068

GENSCREEN FOR MS-COBOL
Cobol Source Code Generator for generating the screen section and data division cobol source code for Microsoft and IBM PC cobol. Screen Image Text files are run through GENSCREEN to produce all of the source code for your screen in less than a minute. Super fast programmer productivity tool $69.99.
Personal Computer Development Corporation
P.O. Box 8556
Warwick, R.I. 02888-8556
(401) 333-8704

FASTSCREEN FOR TURBO PASCAL
FASTSCREEN adds fast screen output capability to your Turbo Pascal programs. Display a full screen or window almost instantly. Read a full screen or window containing multiple input fields with a single procedure call. FASTSCREEN makes it easy to give your programs impressive speed and a clean, professional look. Inline assembler and Pascal routines, all source included. IBM PC/XT/AT and DOS. Color and monochrome. $29.95 from Technisoft
Technisoft
1710 Allied Street Suite 37
Charlottesville, VA 22901
804-979-6464

TECH BOOK

ScreenWorks™ $50
Awesome screen management! Full screen panel editor & designer. Design your most complex screen in 10 min. Supports C & Turbo Pascal. Screen handler written assembler as BIOS ext'n; 50+ library routines give full control of screen, fields, colors & keybd. Developer license avail. 256K, all monitors, no snow. 30 day $back guarantee, $50 chk/VISA/MC.
Keep It Simple Software, Inc.
211 East 43rd Street
New York, NY 10017
(800) 848-8909

TURBO PASCAL GENERATOR
GTP APPLICATION DEVELOPMENT SYSTEM, version 2. Builds complete, working application systems. You give it spec's, it writes error-free code.

Indexed Data Bases	Multiple Screens
Context Sensitive Help	Memory-mapped Video
Global/Phonetic Search	Full Keyboard Supt

Easy to Use Price $150.00 Visa/MC, ck, MO
AEF
P.O. Box 928
Katy, TX 77492
(713) 391-8570

SCREENWARE UTILITIES
FAST ELEGANT SCREEN CREATIONS FOR PC ASSEMBLY ROUTINES. Make new or edit existing screens, even those CAPTURED in OTHER pgms BY OUR PGM! Supports ALL color/char/att & monitors. Pop-up menus, help & file management. Creates binary & assembler-ready ASCII source files. Easy to link to main pgm & invokes w/little code. KBD template, charts, 50pg Man, PC/AT-128K, ONLY $27!
Crosswinds Software
8621 Windjammer Drive, Suite M-56
Raleigh, NC 27609
(919) 847-1812

FORTRAN LIBRARY
A library of over 70 assembly language subroutines complementing MS Fortran compiler. Full range of powerful features, increasing the scope of programming possibilities. Control screen, keyboard, communications; add new language commands. Library includes subroutines for total control of HP LaserJet printer. Institutional purchase orders $99.95, personal checks $49.95 s/h $5.
Psychometric Associates
3434 West Tulsa Street
Chandler, Arizona 85226

SOURCE DEBUGGER FOR LATTICE C
The MSD DeBugger™ is the last, and perhaps final, word in programming assistance for Lattice C users. C Debugger produces a high-level view of C programs via function names line numbers, variable names and C data types, plus a low-level view of machine addresses and instructions for testing assembler language functions. More features include:
- All documentation is prepared for programmers.
- Online help screen throughout the process.
- Capability to single step through your program.
- Set break points, examine registers and variables.

$165.00 + $3.50 shipping VISA/MC

Cdebugger
a high level language debugger

MSD
214 1/2 West Main St.
St. Charles, IL 60174
(312) 377-5151

THE—SCREEN GENERATOR
Powerful Screen Designer and Memory Resident Screen Manager eliminates tedious screen management tasks from your program. Easy access to Screen Manager from your application. Uses disk screen libraries. No Royalties, Not Copy Protected. Uses with FORTRAN, C. BASIC, Turbo-PASCAL, BetterBASIC or Asm (Call if not listed). $125
The West Chester Group
P.O. Box 1304
West Chester, PA 19380
(215) 644-4206

ADVANCED MODULA-2 TOOLS
REPERTOIRE: $64. includes 360K of Logitech or ITC Modula-2 source code: screen compiler and fast display system with input checking, help, windows, menus, colors; auto-adapts to hardware. Integrated multiwindow editor; natural-language analyzer. DOS & BIOS tools. FREE documentation & demo disk. MC/VISA
PMI
4536 SE 50th
Portland, OR 97206 (503) 293-7706
MCI Mail: PMI; Compuserve: 74706,262

WORK HORSE BY USERSOFT
The BASIC code generator that provides:
1) screen painting 2) help screens
3) structured programs 4) maintainable code
5) cuts programming time 6) INKEY$ routine
7) merging 8) printing 9) cross-references
10) runs interpretively or can be compiled.
All this and more for just $59.00.
USERSOFT, Inc.
P.O. Box 517
Manteo, NC 27954
(919) 473-1970

BASIC PROGRAMMERS
ADD MORE HORSEPOWER TO YOUR PROGRAMS!! Use THE BASIC ENGINE™ to build YOUR programs in FAST compiled BASIC. Add your modules to a working core of assembly routines, menus, data screens, pop-up help, sort, more; over 60 subroutines. Complete source for two working programs, full instructions, $79.95. Requires DOS 2+, MS-QuickBASIC compiler.
Praxis Software Engineering Associates
370 Andrews St.
Green River, WY 82935-4846
(307) 875-9626

THE PL/M CONNECTION
- Develop PC software with your PL/M compiler.
- Complete Interface to PC-DOS and PC-BIOS
- Fast Graphics Routines example programs
- All source files are included
- 200 page Technical Reference Manual
- $295.00 No Royalties.

CompuFirm Corporation
7841 Balboa Ave #108
San Diego, CA 92111
(619) 571-0228

PRODUCTIVITY TOOLS
SRMS™ Software Revision Management System stores all versions of source code in a single library. Allows retrieval of any version of source and application of changes while recording when, why, and where changes were made with no duplication of common code. DOS pathname, directory, and environment variable support, typeset manual, much more.
New version (2.0)................................ $125.00.
QMAKE™ is an intelligent system builder patterned after the UNIX make utility only compiles those routines that have changed since last build. Support for macros, multiple entry points, command line parameters. Integrates fully with SRMS™.. $99.00.
MS/PC-DOS 2.0 (MN plus 6%) MC/VISA
QUILT™ COMPUTING
7048 Stratford Rd.
Woodbury, Minnesota 55125
(612) 739-4560

Better BASIC Programmers
BetterTOOLS 2.00 SPEEDS BetterBASIC 1.1/2.0 development. Includes: 190+ useful tools in 17 modules, manual, and source code. No royalties. Quicksort, screen builder, extended math, disk directories, display and printer routines, powerful input, data encryption, on-line error descriptions, much more. Only $89. VISA, MC, COD.
SOFTWARE ASSOCIATES
6220 W. Airport Blvd.
Houston, TX 77035
(713) 726-0706

SOFTWARE/EDIT

Lil'Jake
"The WORLD'S GREATEST Editor." That's what our users say about Lil'Jake.
*Edit multiple files
*DOS commands
*Comprehensive (terse/verbose) online help
*Programming support BASIC, DOS & BIOS
*MACRO facility
We're so sure you'll like it, we want to send you a copy for a free 60-day evaluation. $49.95 + s/h To order: Write or CALL NOW!
Fooks Matthewson Company
P.O. Box 2424 (Oper-28)
Arlington, VA 22202
(703) 684-8284

SOFTWARE/ ENGINEERING

smARTWORK®
Printed-circuit artwork editor for double-sided boards up to 10" by 16", runs on an IBM-PC. Color display allows complete interactive control over the placement and routing process. 2X artwork can be made on a dot-matrix printer or pen-and-ink plotter, $895. Write or call for a brochure.

WINTEK

WINTEK Corporation
1801 South Street
Lafayette, IN 47904-2993
(317) 742-8428

METAL FABRICATORS
PC/Cutlist takes input from your bill of material—Detail drawing and calculates the best cutting combination for any length stock and prints a shop ready cutting list and scrap report. Also an optimization feature finds best mult length for mill orders. Price $300. Demo Disk $25.00
THE JOSEPH ALBERT CO.
P.O. Box 611
Blue Island, Illinois 60406
(312) 349-9032

PC/8087 ARRAY PROCESSING
The VECTOR87 library is written in assembler, includes 60 subroutines to speed scientific programs by 2 to 18x. FFT, vector operations, convolution, etc. 1K real FFT takes 1.2 second. Also Fortran 2D FFT, linear equation solvers. Versions for Lattice C, MS C, IBM Pro Fortran, MS Fortran. $120 each w/source. Check/PO/VISA.
VECTORPLEX Data Systems Ltd.
P.O. Box 138 Station M
Calgary, Alberta, Canada T2P 2H6
(403) 248-1250

TECH BOOK

CREATE PCB's EASILY
Easily create double-sided printed circuit board artwork of up to 21" square using PC-Layout on an IBM-PC. the artwork is created using a color artwork editor. the final artwork, solder mask and silk-screen (text) is produced using either a printer or a plotter. Price $745. Demo disk/manual $45. Demo disk credited to final purchase.
Draco Technology
7210 Jordan Ave., Suite D50
Canoga Park, CA 91303
(800) 235-6646 ext. 646 (outside CA)
(800) 235-6647 ext. 646 (CA only)

SOFTWARE/FINANCIAL

IRRworks™ $25
Are YOU using a spreadsheet for IRR's? Full screen menu and pop-up windows make IRR & PV simple. Fast computation, up to 360 uneven periods. Easy input: steps, % growth, replication. FRACTIONAL PERIODS! Automatic rate conversion. Name & save cases; simple case management. 256K, color/mono, 30 day $ back guarantee, $25 chk/VISA/MC.
Keep It Simple Software, Inc
211 East 43rd Street
New York, NY 10017
(800) 848-8909

HomeWorks $25
Home-buyers, realtors, lawyers, bankers: Compute mortgage payments & totals, any payment period. DETERMINE $$$ YOU CAN BORROW. Print detailed amortization schedules & yearly totals. Name & save cases; simple case mgt. Full screen, windows, help, fast, 196K, color/mono, 30 day $back guar'tee, $25 chk/MC/VISA.
Keep It Simple Software, Inc.
211 East 43rd Street
New York, N.Y. 10017
(800) 848-8909

FINANCIAL CALCULATOR
Memory resident software for PC/XT/AT or compatible computers. Non-resident version also available. Loan amortization, internal rate of return, IRA, real estate, savings, depreciation, and much more!!! FREE shipping in the U.S. 30 day money back guarantee.
NOT COPY PROTECTED. $49.
OPUS Engineering
5825 Vargas Ct.
San Jose, CA 95120
(408) 997-1503

SOFTWARE/GENERAL

PUBLIC DOMAIN SOFTWARE IN C
Over 60 volumes of public domain software in CP/M & MS-DOS formats.
- editors & compilers
- text formatters
- communications packages
- many UNIX-like tools

Write or call for more details.

C Users' Group

THE C USERS' GROUP
P.O. Box 97
McPherson, KS 67460
(316) 241-1065

SOFTWARE/GRAPHICS

TEKTRONIX 4010 EMULATION
High resolution screen/printer graphics for the IBM PC. Full interactive capabilities with file transfer and cross-hair control. Utilizes PLOT 10, protocols. OFF-line review of graphics output. Supports IBM color/graphic or Hercules cards. Easy to use 4010 emulation at an affordable price. $80.00. Demo disk $5.
Technological Systems Group
5044 Haley Court
Lilburn, GA 30247
(404) 923-4980

PLOTTER SOFTWARE
Presentation aides (word charts) are quick and easy to prepare using the DGI SIGNMAKER. With the DGI SIGNMAKER, your PC and plotter (IBM, HP, HI, Epson) your overheads, text slides or signs will have that professional look. $75. For additional information, ordering or to check compatibility contact:
Decision Graphics, Inc.
P.O. Box 2776
Littleton, CO 80161
303-796-0341

MetaWINDOW™
Advanced graphics toolkit provides Xerox Star/Apple MacIntosh style graphics on your IBM PC. Supports most popular graphics cards. Allows you to create pop-up menus, windows & icons; use proportionally spaced fonts; rubberband & drag lines, text or bitmap images; supports mouse-cursor tracking. Tightly optimized for use with Turbo Pascal, IBM Pascal, C, Fortran & Compiled Basic
METAGRAPHICS SOFTWARE CORP.
4574 Scotts Valley Drive
Scotts Valley, CA 95066
(408) 438-5352

TERMINAL EMULATION
For under $100, MicroPlot's PC-Plot III graphics emulator software package will enable your IBM PC and compatibles to appear to a mainframe as a DEC VT-100/VT-52, a Retrographics VT-640, a Tektronix 4010/4014 or a partial Tektronix 4027. Call 1-800-338-0333, In Ohio 1-800-242-0333. Operator ID.: 766-8501
MicroPlot
659-H Park Meadow Road
Westerville, OH 43081
(614) 882-4786

SCIENTIFIC DATA PLOTTING
SCI-GRAF creates graphs up to 1680 x 1712 dots (over 3 million pixels!) on Epson or IBM graphics printers. Supports log scaling, overlays, point-labeling, legend creation, batch mode, wide-carriage printers, and color graphs on a JX-80. Requires DOS 2 or 3, 256K. No credit cards. $99.95
Microcomputer Systems Consultants (MSC)
27-P Forest Avenue
Port Jefferson Station, NY 11776
(516) 928-7493

SOFTWARE/LANGUAGES

EASY WAY TO MASTER DOS!
New handbook shows how, why, and when to use all 44 PC & MS DOS commands quickly and easily! Lively hands-on tutorial makes learning fun, for a change. Covers basic thru advanced topics. All versions up to 3.1. Lots of practical examples. 221 pages only $15.95 at bookstores, or send check + $1.00 S&H to:
LITTLE, BROWN & COMPANY PUBLISHERS
200 West Street, Code T-368121
Waltham, MA 02254
1-800-343-9204 (credit cards)

FREE 8088 TUTOR Monitor
Learn 8088 assembly language using the new book, IBM PC-8088 Assembly Language Programming, by Richard E. Haskell. Order the book for $25 plus $3 s/h and get the powerful TUTOR monitor software free. Load and examine any disk file, disassemble any 8088 code, debug EXE files. Satisfaction guaranteed.
REHI Books
54 N. Adams, Suite 130-T
Rochester Hills, MI 48063
(313) 853-1941

SOFTWARE/ONE OF A KIND

INTRODUCING-JORJ
JORJ- The first and only phonetic dictionary! 50K words/15K definitions. Finds words super fast for both good/poor spellers. JOSE a fun word-game is included in iinitial release as a bonus. Requires IBM-PC DOS2 & Hard disk. Copy protected. Price $100. JORJ demo disk $10. Check/M.O. Mich. residents add 4% sales tax.
JORJ SOFTWARE CO.
4354 Fletcher Rd.
Manchester, MI 48158
(313) 428-8010

SOFTWARE/OPERATING SYSTEMS

UN/EMULATOR BY WAWA
Run CPM on your PC or clone at MACHINE SPEED. UN/EMULATOR runs native 8080 code and increases DOS performance by 10%. We include disk conversion software, terminal emulators, and serial transfer utility. Available in 5 MHz and 8 MHz. Starting at $99.95 plus $5 S&H. Dealer inquiries welcome. Call or write:
WEITZMAN AND WOOD ASSOCIATES
580 N.W. 99th Way
Pembroke Pines, FL 33024
(305) 432-8025

SOFTWARE/SCIENTIFIC

SPARSE/DENSE MATRIX
MATRIX: Does to matrix what TK! solver did for numbers. Solves lin/nonlin/diff eqns. Programmable Matrix functions. $59.95
SPARSEPACK: Turbo Pascal source code for user-transparent Sparse Matrix handling. Overcome 64K barrier and more. $49.95. VISA/MC call 1-800-835-2246 ext. 80.
SoftTech Inc.
18505 W. 8 mile.
Detroit, MI 48219
1-313-544-8544

SOFTWARE/SECURITY

SECURE AT/XT/PC
Control system access, data access! FiXT/S. Control system boot for most popular XT/PC hard disk controllers. Vfeature for AT-and-XT-compatible HD controllers segments hard disk by volumes, controls access with passwords, supports hard disk expansion. $80 $120 + $3 shpg. plus CA tax.

Golden Bow Systems
2870 Fifth Ave. Suite 201
San Diego, CA 92103
(619) 298-9349

BITLOCK® SECURITY
Proven effectiveness of powerful multilayered security on IBM PC/XT/AT Compatibles. TRS-80, Apple & Commodore computers using rapid decryption algorithms. Total transarency after easy attachment of small reliable security device to external port. Inexpensive, yet has survived piracy attempts > 3½ years.
MICROCOMPUTER APPLICATIONS
7805 S. Windmere Circle
Littleton, CO 80120
(303) 798-7683 or 922-6410

TECH BOOK

SMART COPY PROTECTION
Attention Software Developers:
In the past, Copy-Protecting your program or system meant inconvenience to the end-user, complicated production techniques and a lot of added expense per product copy. Our new EVERLOCK product has solved these problems. For IBM PC/XT/AT and compatibles. Call us.
Az-Tech Software, Inc.
426 Grandview
Richmond, MO 64085
(816) 776-8153

SIDEVIEW: PC/AT MENU
Password protect your hard disk today! DOS 3.0/3.1 hard disk manager. Allows you to create color menus to run any program or DOS command. Prevent hard disk erasure. Works with XT's too. 25 pg manual. Money back guarantee. PO's & phone orders accepted. $39 + $3 shpg.
Keller Software
1825 Westcliff Drive
Newport Beach, CA 92660
714-642-5544

SOFTWARE/SERVICES

TAPE/DISK CONVERSIONS
Conversion services to or from over 500 computer systems:
- Magtapes
- Micro Computers
- Mini Computers
- Word Processors
- Typesetters

Our conversion capabilities surpass most in the industry.
Pivar Computing Services, Inc.
165 Arlington Hgts. Rd. #T
Buffalo Grove, IL 60089
(312) 459-6010

TAPE AND DISK CONVERSIONS
TO/FROM cards, disk, tape ASCII/EBCDIC specializing in IBM-PC/XT/AT and compatibles. Reasonable prices, satisfaction guaranteed. Total Data Processing services with in-house mainframe.
D.P. Modern Systems
804 N. Reeder Ave.
Covina, CA 91724
(818) 966-2628

SOFTWARE/STATISTICS

RATS!
RATS is the ultimate econometric software package. It is powerful, fast, accurate, and inexpensive. RATS performs regression analysis, time-series forecasting (including Box-Jenkins) and cross-section (including logit and probit). RATS supports a variety of data formats, including Lotus WKS files. RATS also offers a programming language that provides maximum flexibility and power. Spectral analysis and graphics are available as options. Base program $200. Demo including 300 page manual $40. MC/VISA.
VAR Econometrics
P.O. Box 19334
Minneapolis, MN 55419
(612) 822-9690

ELF—THE STATISTICAL PKG.
Interface with Dbase and Lotus files or will create its own. Factor analysis, stepwise regression, discriminant analysis, 1&2 way anova crosstabs, correlations, t-test, frequencies, transformations .... unlimited observations, 500 variables in database. All new and improved manual. 30-day examination. Call for more.
The Winchendon Group, Inc.
P.O. Box 10339
Alexandria, VA 22310
(703) 960-2587

SOFTWARE/TERMINAL EMULATION

VT-102 & 4010 EMULATION
PC 102 turns an IBM PC/XT/AT/jr to a VT-102/100/52 terminal with printer/file transfer support. Optional: 132 col., 4010 graphics support. Guaranteed compatibility with all VT-100/52 applications including EDT, WORD II, All-In-One, & UNIX. Easy to use. 3000+ sold. From $89. MC/VISA/COD. Call for free package! Moneyback guarantee.
General Micro Systems
P.O. Box 5330
Hopkins, MN 55343
(612) 944-0593

SOFTWARE/UTILITIES

AT/XT/PC HARD DISK EXPANSION
"Replace hard disk with a bigger one, or add a second drive! Vfeature BREAKS THE 33 MBYTE BARRIER on standard AT, XT, and compatible hard disk controllers. Includes multiple volumes, security features, selectable clusters, keyboard lock. $80-$120 + $3 shipping + CA Tax"

Golden Bow Systems
2870 Fifth Avenue, Suite 201
San Diego, CA 92103
(619) 298-9349

DISK MECHANIC
THE ULTIMATE Floppy Disk Backup & Repair Utility. Can back up ALL software protected disks written on the IBM PC. Works manually or automatically. Files or sectors can be restored, searched, examined & changed. Checks disk drive speed, req. IBM PC, XT, AT, DOS 1, 2, 3, 192K + 64K if only 1 floppy drive. $73 ppd. USA
MLI MICROSYSTEMS
PO BOX 825, Dept TB2
Framingham, MA 01701 USA
(617) 926-2055 for info MC/VISA

CHARACTER CUSTOMIZATION
CHARGENI 2.1 works with the IBM EGA to let you modify the character set, allowing many word processors to display technical material, equations or other special characters. Also works on the Color/Graphics Adapter in graphics modes. Requres DOS 2.x or 3.x, IBM Standard or Enhanced Graphics Adapter. $35 + $2 s/h (MN add 6%).
DK Micro Consultants
P.O. Box 6714
Minneapolis, MN 55406
(612) 722-0931

MenuWorks™ $25
Using a PC becomes as easy as turning it on: Start up all programs from menu. Customize selection screen. Just Tab & Enter to Run. Prompts for input if needed. Super slick editor to set up menu; easy to change. Fast. Full screen, windows, help, 196K, color/mono, 30 day $ back guarantee, $25 chk/VISA/MC.
Keep It Simple Software, Inc.
211 East 43rd Street
New York, NY 10017
(800) 848-8909

dirWorks™ $25
PC users: avoid DOS drudgery! Complete disk mgt made simple: copy, erase, rename, format, all directory commands and TONS more. Multiple file & directory operations, Sorts display by time, date & alpha. Edit files without going to DOS. Full screen, windows, help. Fast, 196K, color/mono, novice & expert mode, 30 day $ back guarantee, $25 chk/VISA/MC.
Keep It Simple Software
211 East 43rd Street
New York, NY 10017
(800) 848-8909

JetWorks™ $25
Attention HP LaserJet users: Control LaserJet & forget escape sequences forever. Full menu and window selection for all options & cartridges. Creates batch files to avoid repetitive setups. Prefab font files for all current cartridges. Full screen, windows, help, fast & simple, 196K, color/mono, 30 day $back guarantee, $25 chk/VISA/MC.
Keep It Simple Software, Inc.
New York, NY 10017
(800) 848-8909

PC-COMPANION
Menu subsystem, a must for any PC with a hard disk supporting multiple applications or users. Single key stroke execution of software. 15 entries per menu. Set drive, path, directory, printer. Password access to menus. Supports all types of command execution Command logging. Screen colors definable. Written in assembler for rapid access. Requires 32k MSDOS/PCDOS 2.0 or higher. $39.95 (TX resident add 6.125% tax).
BINARY ENGINEERING
12841 Jones Rd. Suite 101
Houston, TX 77070
(713) 995-3770 VISA/MC

TallScreen—DOS POWER
Natural extension of DOS. Scroll back through screen output, edit text on full screen, mark blocks to printer or file, recall commands & directories, enter multiple commands, capture screens from application programs, create user profiles. Solid tech support. A real bargain at $49.95 VISA/MC

QUALITAS™
Qualitas, Inc.
8314 Thoreau Drive
Béthesda, MD 20817
(301) 369-8848

EXTENDED FILE UTILITIES
Replace MS-DOS COPY, DELETE, RENAME with these fast, powerful utilities (DOS 2.0 or later). File lists with commas, optional confirmation querying, access to hidden/system files, ASCII-mode copying. Just $15 + $1.95 s/h (MA res. add .75 tax), MC/VISA OK.
Pocumtuck Valley Software
P.O. Box 156-A
South Deerfield, MA 01373

EFFICIENT DOS
PCED: Use DOS more efficiently without menus, batch files, keyboard enhances. Command editor with recall; synonyms execute one or many commands with batch-style variables; automatic parameter recall; command logging; MUCH more. Very flexible. Recommended by Doctor Dobbs and CPCUG. $35 plus $4 s/h; VISA/MC.
The Cove Software Group
P.O. Box 1072
Columbia, MD 21044
(301) 992-9371

CONCURRENT DOS BACKUP
BackPack™ runs like BACKUP/RESTORE and supports DOS and CP/M media as well as users. Backs up and restores up to 1000K/min. on an AT and 360K/min. on an XT. Supports release 4.1. Also available on CP/M-80 and CP/M plus. $150.
Bright Light, Inc.
520 Fellowship Rd. #C301
Mt. Laurel, NJ 08054
(609) 778-0772

TECH BOOK

Directory eXtended™

BE IN COMMAND! New version of very popular command-oriented file manager! POWERFUL batch file generation-perform any task! FLEXIBLE file matching and sorting! CONTROL file attributes! TEST file integrity!. MONITOR disk params; 2-col display! MUCH MORE! Send $35+$5 p&h (in TX, add tax); money back guarantee!

DIRECTORY eXTENDED(TM)

ECONO-SOFT Dept. T66
9200 North Plaza # 1906
Austin, TX 78753

HELL uva SHELL

Most DOS interfaces are designed for IDIOTS! Get the BEST full-screen interactive directory around ... DOS command recall ... command aliasing ... multiple commands at DOS prompt ... find files anywhere on disk ... find complex text patterns in files ... browse files ... copy files based on time stamp & MORE. Only $39! DOS 2.0+, uses 96K memory. NCP!

Dwarf Nebula Software
666-SHELL
P.O. Box 46, Dept. TJ
Sugarland, TX 77487

EASYINDXE PLUS™

Cross reference all source code by file name. Create index by page number. Unlimited number of indexes and accessed files. Word or phrase indexes up to 59 characters. Table of contents generation, typesetting, and print formatting. Supports over 100 printers. $39.95 $3 shipping (CA. RES. ADD $2.60 Tax)

Foreman Software
977 Clayton St. D4
San Francisco, CA 94117
(415) 665-6414

TurboRef 4.0

Pascal programmers get organized with TurboRef! TurboRef will cross reference a source program and create a program listing. TurboRef can process a list of files, will read "include" files, identifies line number for each reference. New release now avail. IBM PC and jr., XT, AT or compatible, only $49.95; VISA/MC, or check.

GRACON SERVICES, INC.

GRACON Services Inc.
4632 Okemos Rd.
Okemos, MI 48864
(517) 349-4900

TURBO PACKAGE for Turbo Pascal

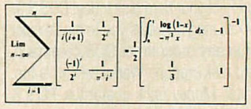
*Crash the 64K barrier. Forget MODULA2.
*1000 times faster than chains or overlays.
*Precompiler turns short calls into long calls. Run 500K of memory resident code.
*32 bit integer math toolbox, ASM.
*Mix Turbo, TurboBCD, and Turbo-87.
*Free information.
*Turbo Package plus tools, $39.95.

CONVERSATIONAL COMPUTER SYSTEMS
5371 Verbena Road
San Antonio, TX 78240
(512) 692-0353

SOFTWARE/WORD PROCESSING

FORMATH™ TEXT-FORMATTER

For technical texts. Sizes & places equations, matrices, ratios, integrals, big symbols, footnotes. Macros, multiple fonts, 158 Greek/math symbols. Automatic hyphenation, section/equation/reference numbering, indexes, table of contents. Dot-matrix, daisywheel, laser printers, plain/graphics monitors. $400. $50 for demo.



SHANTHA SOFTWARE, INC.
50 West 97th St. Room 11N
New York City 10025
212-222-7647
Touch-tone Toll-free: 950-1088-wait-FORMATH

INDEX TO ADVERTISERS

PC TECH JOURNAL JUNE 1986

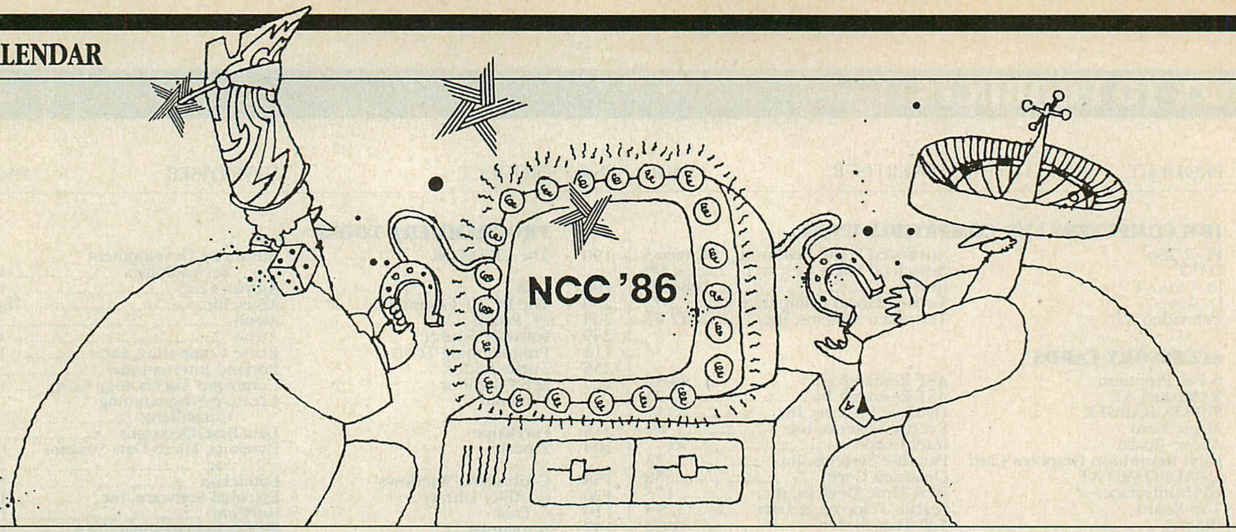


ILLUSTRATION • DAVID POVILAITIS

JUNE

June 3-5
13th International Symposium on Computer Architecture
Tokyo, Japan
Sponsor: IEEE-CS, ACM, and Information Processing Society of Japan
Contact: Shunichi Uchida, ICOT, Mita Kokusai Building 21F, 4-28 Mita 1-Chome, Minato-ku, Tokyo 108, Japan

June 4-6
National Educational Computing Conference
San Diego, CA
Sponsor: ACM, IEEE, SCS
Contact: Susan M. Zgliczynski, University of San Diego, School of Education, Alcala Park, San Diego, CA 92110

June 16-19
National Computer Conference
Las Vegas, NV
Sponsor: AFIPS, ACM, IEEE-CS, DPMA, and SCS
Contact: AFIPS, 1899 Preston White Drive, Reston, VA 22091; 703/620-8900

June 22-25
23rd Design Automation Conference
Las Vegas, NV
Sponsor: ACM SIGDA and IEEE-CS
Contact: J. D. Nash, Raytheon Company, Bedford, MA 01730; 617/274-7100, extension 4758

June 22-August 1
National Computer Camps
CT, OH, GA
Contact: National Computer Camps, Box 585-CA, Orange, CT, 06477; 203/795-9667

June 23-27
Compiler Construction
Palo Alto, CA
Sponsor: ACM
Contact: ACM 11 W. 42nd Street, New York NY 10036; 212/575-1520

JULY

July 8-11
UNIX: A Comprehensive Workshop
Washington, DC
Sponsor: Integrated Computer Systems
Contact: Yolande Amundson, Integrated Computer Systems, 6305 Arizona Place, P.O. Box 45405, Los Angeles, CA 90045; 800/421-8166

July 9-11
PC EXPO
New York, NY
Contact: 333 Sylvan Avenue, Englewood Cliffs, NJ 07632; 201/569-8542

July 21-25
European Conference on Artificial Intelligence
Brighton, England
Sponsor: European Coordinating Committee on Artificial Intelligence and the Society for the Study of Artificial Intelligence and Simulation Behavior
Contact: Benedict du Boulay, ECAI, The University of Sussex, Cognitive Studies Programme, Brighton BN1 9QN, U.K.

July 21-26
Third International Conference on Logic Programming
London, England
Contact: Doug DeGroot, IBM Research, P.O. Box 218, Yorktown Heights, NY 10598

July 22-25
Ada Programming Workshop
Washington, DC
Sponsor: Integrated Computer Systems
Contact: Yolande Amundson, Integrated Computer Systems, 6305 Arizona Place, P.O. Box 45405, Los Angeles, CA 90045; 800/421-8166

AUGUST

August 4-6
LISP and Functional Programming
Cambridge, MA
Sponsor: Association for Computing Machinery
Contact: Richard P. Gabriel, Lucid, Inc., 707 Laurel Street, Menlo Park, CA 94025

August 5-7
ACM SIGCOMM Futures in Communications Conference
Stowe, VT
Contact: Walter Kosinsky, Norwich University, Northfield, VT 05663; 802/485-5011, ext. 237

August 11-13
Symposium on Principles of distributed Computing
Calgary, Alberta, Canada
Sponsor: ACM SIGACT-SIGOPS
Contact: ACM, 11 West 42nd Street, New York, NY 10036

August 18-22
13th Annual Conference on Computer Graphics and Interactive Techniques
Dallas, TX
Contact: SIGGRAPH '86 Conference Management, 111 East Wacker Drive, #600, Chicago, IL 60601; 312/644-6610

August 19-22
1986 International Conference on Parallel Processing
St. Charles, IL
Sponsor: ACM
Contact: IEEE-CS, 1730 Massachusetts Avenue, NW, Washington, DC 20036-1903

SEPTEMBER

September 1-6
World Computer Conference
Dublin, Ireland
Sponsor: International Federation for Information Processing
Contact: IFIP Congress '86, 1899 Preston White Drive, Reston, VA 22091; 703/620-8900

September 8-10
NCC-Telecommunications Conference
Philadelphia, PA
Sponsor: AFIPS, ACM, DPMA, IEEE, SCS
Contact: Mike Sherman, AFIPS, 1899 Preston White Drive, Reston, VA 22091; 703/620-8935


EXTEND
YOUR
COMPUTING
POWER